UNIVERSITY OF CANTERBURY

Department of Mechanical Engineering

Christchurch New Zealand

Control of Nitinol Wire Actuator

by

Angelo Garcia

A thesis submitted in partial fulfilment of the
requirements for the Degree
of
Master of Engineering
at the University of Canterbury

November 2003

Abstract

The Stewart Platform is well known parallel robotic mechanism often used for flight simulators. Less well known applications involve robotics and machine tools. This research deals with Stewart Platforms that use Nitinol wire, a shape memory alloy, as binary actuators that are switched on and off to control the position of the platform. Binary control of the six actuators on a platform yields sixty four possible platform positions, and four stacked platforms yield $256E^{12}$ possible positions. The design of a simple basic four stages model is described in this research.

Experiments were conducted on the Nitinol wire to investigate the shape memory effects, hysteresis and the martensite/austenite phase transformations. These results were used in the design of a high power to weight ratio Stewart Platform, overcoming many of the drawbacks of previous models that are powered by heavy hydraulics or electric motors. Due to the complexity of some of the parts and small numbers required, rapid prototyping was used to manufacture over 90% of the structure. In this research, the control system proposed is expected to be useful for not only Shape Memory Alloy (SMA) actuators but also other actuators with hysteresis, for example, piezoceramic actuators, rubber actuators driven by air supply, bellows actuators, etc.

Acknowledgements

There are always many unseen contributions to a thesis report. First of all, I would like to thank my supervisor Dr. Reg Dunlop and Dr. John Smaill whose ideas and advice have made this thesis possible. Also to Dr Ilanko for the derivation of the curved beam.

Julian Murphy, Julian Philips, Rodney Elliot, Andy Cree, Graham Harris and Ron Tinker of the Department of mechanical Engineering technical staff for their advise, expertise and direct involvement in various construction phases.

Shigeaki Kinoshita and Eugene Chin for their invaluable proof reading.

My parents, Stanley Garcia and Theresa Kam, have contributed immensely to my life and during the time of this thesis was no exception. Thanks also go to all of my friends who helped to maintain my sanity during these academic years, especially

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Brief Description of Thesis

The aim of this research is to design and build a multi-degree of freedom nitinol actuated mechanism. The Nickel Titanium shape memory alloy (SMA) was developed at the Naval Ordnance Laboratory. It represents a new class of material, capable of transforming thermal energy into mechanical work. It has a chemical composition in the range from about 53% to 57% by weight of nickel, with the remainder being titanium. The name "shape memory alloy" refers to its ability to restore to its original shape even after it has been deformed. The return to the alloy's original shape requires a certain amount of heat input, because this alloy exists in different phases at different temperatures. At low temperature, the alloy has a martensitic structure whereas at high temperature the alloy is austenitic. The restoring process allows the nitinol to change phases from a martensitic structure to the original austenite phase. In between these two phases, there is a temperature range over which austenite and martensite co-exist. Furthermore hysteresis is associated with the martensitic transformations; i.e. the transformation temperature differs upon heating and cooling. If direct-pulsed electrical current controls the heating and cooling of these nitinol wires, repeated cyclic motions can be achieved. Thus it can be used for thermal-mechanical actuators.

Nitinol wire was chosen for use as an actuator, not only because of the thermal-mechanical properties, but also it is much simpler in operation than electric motors or hydraulics arms. Moreover, they are characterized by high power/weight ratios, good corrosion resistance and strength. Thus they are particularly well suited for operation in extreme environments such as might be found in space, radioactive environments and in the deep sea; in these cases the control and maintenance of motors and hydraulic units are very difficult (largely due to the corrosion effects). The specific actuator proposed here is designed to demonstrate the utilisation of nitinol wire as the actuator.

The basic design idea is that the mechanical nitinol-actuated mechanism should be able to perform complex motion. This means that it can alter its shape of its own accord over its whole length. With this mobility, the mechanism would be able to penetrate inside environments containing much chaotic piping or the like (e.g. inside nuclear reactors such as Fig. 1.1), and to undertake visual examinations with the television cameras fitted at its ends. Current similar systems are flexible endoscopes, which use fibre-optic bundles for imaging. Endoscope are indispensable for medical examination of internal organs of the body such as the stomach, intestines and trachea, for the inspection of jet engine interiors, and for detailed observation of internal mechanisms of complex machines. The main weakness of present-day endoscopes lies in the fact that only their end section can freely describe a

curve (see Fig 1.2). For this reason, depending on the use, instances occur in which it is difficult to move the endoscope's end section to the region to be inspected. To compensate for this difficulty a special technique is required for the insertion of the endoscope; it is called hook insertion method. In this method the endoscope is twisted as well as bent during the insertion, to negotiate the curvature of the path for the working region. This mechanism has the disadvantage of not being able to transmit the force to the tip when the endoscope is curved in a complex manner. For this reason the idea of Stewart Platforms was utilised to the design.

At present several three dimensional manipulators and walking robots prototypes based on nitinol actuator have been manufactured by universities and industries e.g. Stiquito from Indiana University [1] and manganese nodule mining system from the Japanese industrial agency [2] . These robots have been aggressively developed in Japan during the last 5 years. However, there are many technical issues to be resolved such as energy efficiency in transforming electrical power into motions and the response speed of the actuator has to be optimised. The object of this research is to extend the utilisation of nitinol actuators further into industrial fields.

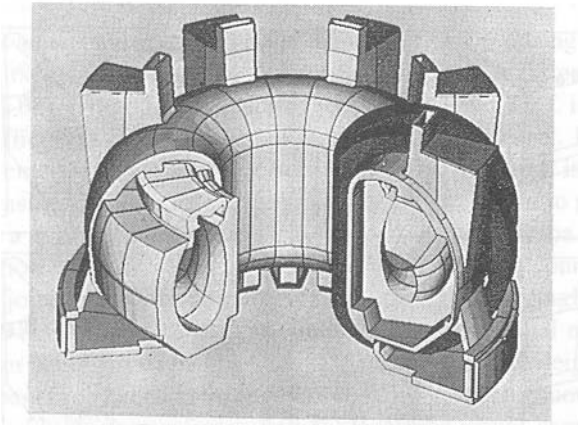


Figure 1-1 Model of the interior of a nuclear fusion reactor [3].

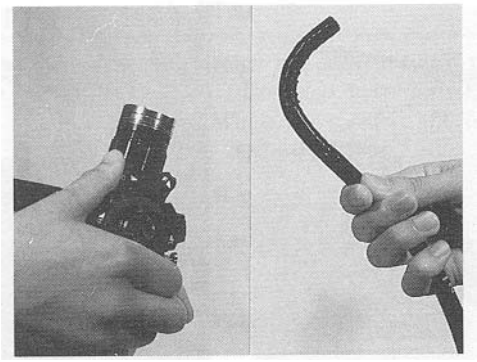


Figure 1-2 An example of a colonoscope [3].

The basic arrangement for the nitinol wire actuated mechanism consists of multi-decked identical prismatic triangular box arranged as multi-stage Stewart Platforms such as shown in Fig 1.3. Every stages of the Stewart Platform were connected and supported by the leaf springs. Motors or hydraulic actuators are unnecessary. The preloading leaf springs would also keep the nitinol wires under tension when relaxed. Each stack of the nitinol actuator has six degree of freedom and a total of 2^6 combinations of movement. Hence the nitinol actuated platform should be capable of performing some complex movements. Thus, a certain degree of rotation and twisting is possible. With this configuration the nitinol actuator would attain a larger working envelope and minimises the danger of pile-up between each nitinol wire. It also eliminated the disadvantage of not being able to transmit the force to the tip when the current design endoscope is curved in a complex manner. Moreover the prototype multistage Stewart Platform produced during this research can be scaled in size to suit the application. Detailed design is to be discussed in chapter 3.

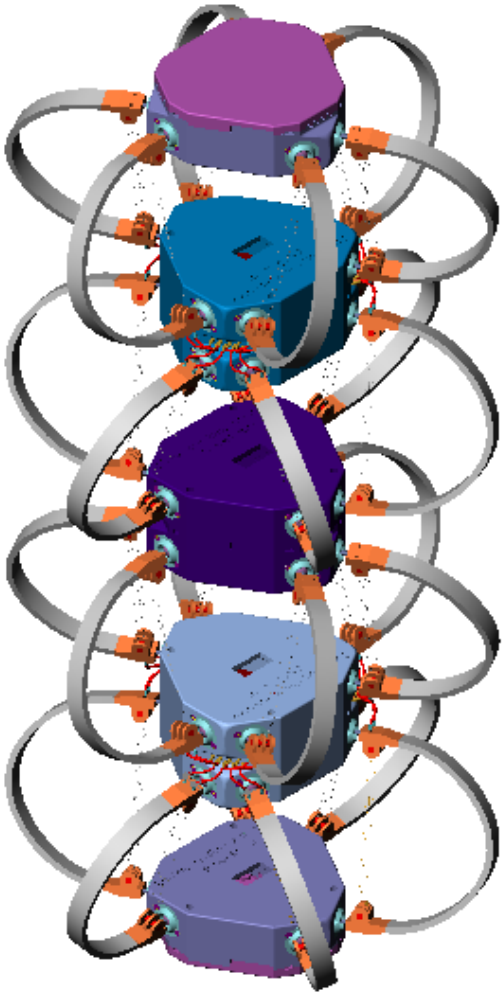


Figure 1-3 Solid modelling of a four stages Stewart Platform.

The work in this thesis has resulted in a snake type robot that is aesthetically, dimensionally and operational quite similar to a snake. There is however some scope for improvement on this prototype design.

1.2 Thesis Outline

The thesis is organised into 8 chapters and 5 appendices. The material presented in this thesis follows a logical progression starting with a brief review of the existing SMA actuated endoscopes. The development work undertaken and the performance of the SMA actuated snake robot are described. A list of bibliographic references to published works supporting the work presented here is included. The contents and scope of individual chapters in this thesis are as follows.

Chapter 2 discuss the properties and behaviour of the SMA. In chapter 3, a brief review of the history of Stewart Platform and the latest manufacturing technique – Rapid Prototyping. This is followed with the design of the nitinol actuated snake robot; force analysis and energy methods are applied to determine a suitable curvature and cross section of the leaf spring. Then in chapter 4 the testing methodology and results are presented. A general method for approximating the changes the changes in position and orientation of the Stewart Platform is proposed in chapter 5. The hardware is described and synergy of the control system is given in chapter 6. Chapter 7 presented the performance data for the first prototype. In the concluding chapter, the results achieved are discussed and the areas for further research are identified.

Chapter 2 Background Theory of NiTi Wire

2.1 One-Way Shape Memory Effect

Nitinol is one of the high-performance shape-memory alloys. These nitinol wires have been specially processed to have large stable amounts of memory strain for many cycles. In other words, they contract like muscles when electrically driven. This ability to shorten is characteristic of certain alloys, which dynamically change their internal structure at some temperature. They contract by several percent of the length when heated and then easily elongate again (using a relatively small load) when the current is turned off and they are allowed to cool. The shape memory effect observed in nitinol is the result of a solid state transformation between two phases: austenite and martensite. See Fig 2.1. The microscopic and macroscopic properties related to this effect are discussed below.

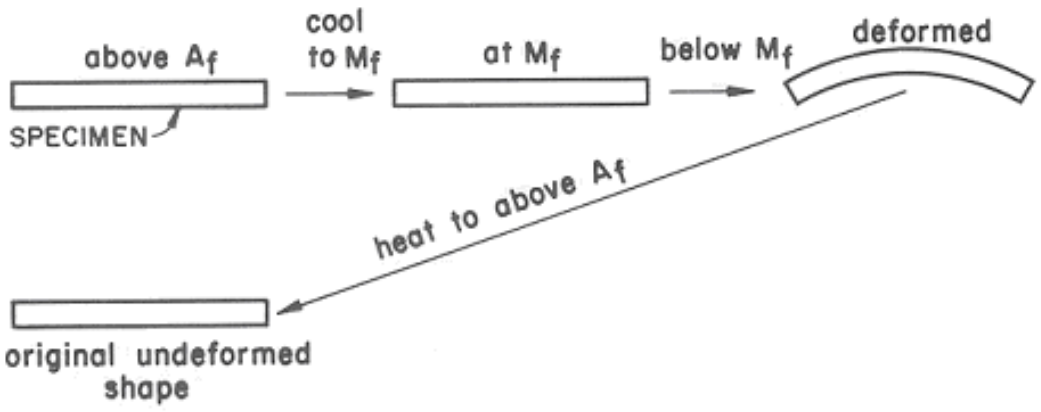


Figure 2-1 Macroscopic One-Way Shape Memory Effect [3].

2.2 Martensite and Austenite

Shape memory alloys exhibit the shape memory effect due to a transition between a parent phase and a product phase, where the term phase refers to a solid structure state of the material.

The parent phase is known as austenite. Austenite is a crystalline structure, usually with body-centered symmetry, or a variant of this symmetry i.e. a cubic structure, with atoms at each vertex and one atom in the center of the cube. If the types of the atoms at each vertex (or the center) are random, the lattice is known as disordered, while if a specific type of atom is always at a specific site in the cubic lattice then it is known as ordered. In steel, the atoms are disordered, while in nickel-titanium alloy (Ni-Ti) the atoms are ordered within the lattice.

The product phase is known as martensite. Martensite has a rhombic structure. (The difference between the structure of austenite and martensite can be likened to the difference between a square and a parallelogram). The two phases can be seen in Fig 2.2.

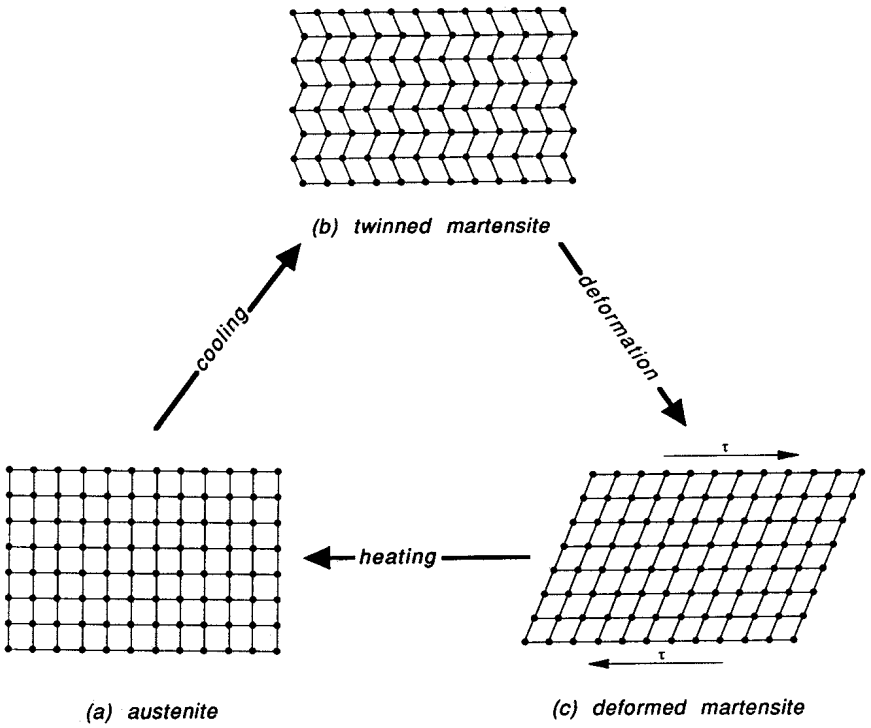


Figure 2-2 Microscopic One-Way Shape Memory Effect [3].

On a macroscopic scale, the SMA material in its product phase is not one continuous sheet of martensite, but a conglomerate of martensite plates. In Ni-Ti alloy these plates are very fine, but in many other SMA the plates are large enough to seen with an optical microscope.

The phase of the material is temperature-dependent. At high temperatures, shape memory alloys such as NiTi are principally austenite while at low temperatures martensite predominates.

2.3 Martensitic Transformation

Transformations between two solid phases, such as austenite to martensite, occur in two ways:

- *Diffusional transformations*
- *Displacive transformations*

Diffusional transformations entail a change in the chemical composition of the solid structure. To cause this change in chemical composition, atoms migrate randomly within the solid over

relatively large distances. Since atomic migration is involved, diffusional transformations depend on the temperature of the solid and time period which the temperature is maintained.

In contrast, displacive transformations do not entail a change in chemical composition. Instead, the crystal structure changes to a more stable structure, by local rearrangement of the atoms. Since no large-scale atomic migration occurs, displacive transformations only depend on temperature. If the crystal structure is absorbs enough energy to change state, the change of state occurs. The time period over which the temperature is maintained is irrelevant.

Martensitic transformations, which involve a change to a martensite crystal structure when higher temperature austenite is cooled, are displacive transformations. Since a more stable crystal structure is being formed, energy in the form of heat is released during the transformation process. Also, there is a temperature range over which both austenite and martensite coexist within the solid. The ordering of atoms within the austenite crystal lattice is maintained in the martensite.

Martensitic transformation occurs in two stages:

- Bain strain
- Lattice-invariant shear

Bain strain involves the atomic movements which cause the austenite structure to be transformed into the martensite crystal structure. Considered in two dimensions, this involves an atomic shift such that the square austenite structure becomes the parallelogram martensite structure. On a larger scale, a lattice of square austenite crystal structures deforms into a lattice of parallelogram structures. As the transformation proceeds row-by-row, at any one time before the transformation is complete, there is an interface between austenite and martensite call the martensite interface. The transformation process is shown in Fig 2.3

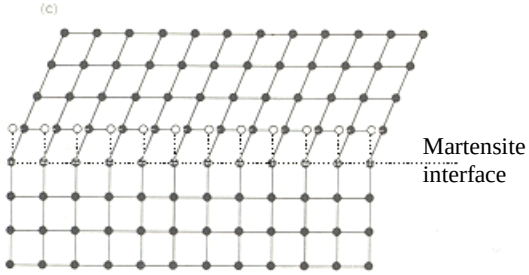


Figure 2-3 Bain Strain – Martensite Interface [3].

Clearly, the shape of the overall lattice is going to change to a large parallelogram. However, such a shape change may not be possible due to the surrounding crystal structure. (Since martensite actually “grows” in plates, the adjacent lattice may still be mostly austenite and thus immobile). Hence, the second motion comes into play: *lattice-invariant shear*. Lattice-invariant shear involves re-organizing of the martensite structure to fit within a constrained volume.

There are two ways in which this can happen:

- Slip
- Twinning

Slip means that each row in the martensite lattice shifts so that the overall lattice maintains a square shape. However, this shifting requires the breaking of atomic bonds and thus is not reversible-permanent deformation.

Twinning is a reversible process in which each row of parallelogram-shape martensite leans the opposite way to neighbouring rows. For example, all the even rows may lean to the right, while all the old rows may lean to the left. Thus the overall structure is square, like the original martensite. However, no atomic bonds have been broken. Hence twinning is a reversible process, and thus forms the basis for the shape memory effect. Twinning is also shown in Fig 2.2

The boundary between rows of parallelograms leaning in opposite directions is called the twin boundary. These boundaries have very low energy and are quite mobile. Thus if an external force or stress is applied to the martensite structure, the twin boundaries will move (without slip) to accommodate it.

If appropriate stresses are applied, the martensite can be coerced into a lattice structure in which all parallelograms lean the same way i.e. one variant of the martensite structure completely predominates. This is called detwinning. Fig 2.2 provides an example of fully-detwinned martensite and twin boundaries.

2.4 Macroscopic Phenomenon

Consider a SMA in the martensitic form. Due to the ability of martensite to exhibit twinning, the martensite can be deformed by external stresses without fracturing the rhombic lattice structure.

However, when the martensitic alloy is heated past a certain temperature (the austenite start temperature A_s) the martensite begins to transform into austenite. After the austenite finish temperature $A_{f,}$ is reached, all the martensite will have been transformed into austenite. Austenite only exists as a cubic lattice. It does not exhibit twinning. Hence it only supports one shape – the "remembered" shape of the alloy. Thus heating the alloy causes it to restore its original, undeformed state (if unconstrained), as shown in Fig 2.4

When the austenite alloy is cooled past the martensite start temperature M_s, martensite begins to form within the alloy. After the temperature is reduced below the martensite finish temperature, M_f, all the austenite has become twinned martensite. This does not entail a change in shape. The martensite is twinned because it was constrained by the surrounding crystal lattice (during the martensite transformation) to maintain the shape of the austenite. However, once all the austenite has been transformed into martensite the material can be deformed anew by application of external stresses. (It can actually be deformed earlier, but the stress required would be greater.)

This temperature induced phase transformation results in a corresponding change in the physical properties of the alloy. The most obvious physical change is the change in shape, usually manifested as a change in strain, or deformation of the alloy. Although the shape only changes during the heating of the alloy past A_s, properties such as the electrical resistance change during both the heating and cooling transformations. This has useful applications for inherent position and force detection. The temperature dependence of electrical resistance is shown in Fig 2.4. The electrical resistance is also dependent upon the applied stress. For an exploration of the hysteresis in the resistance-stress relationship refer to Ikuta [4].

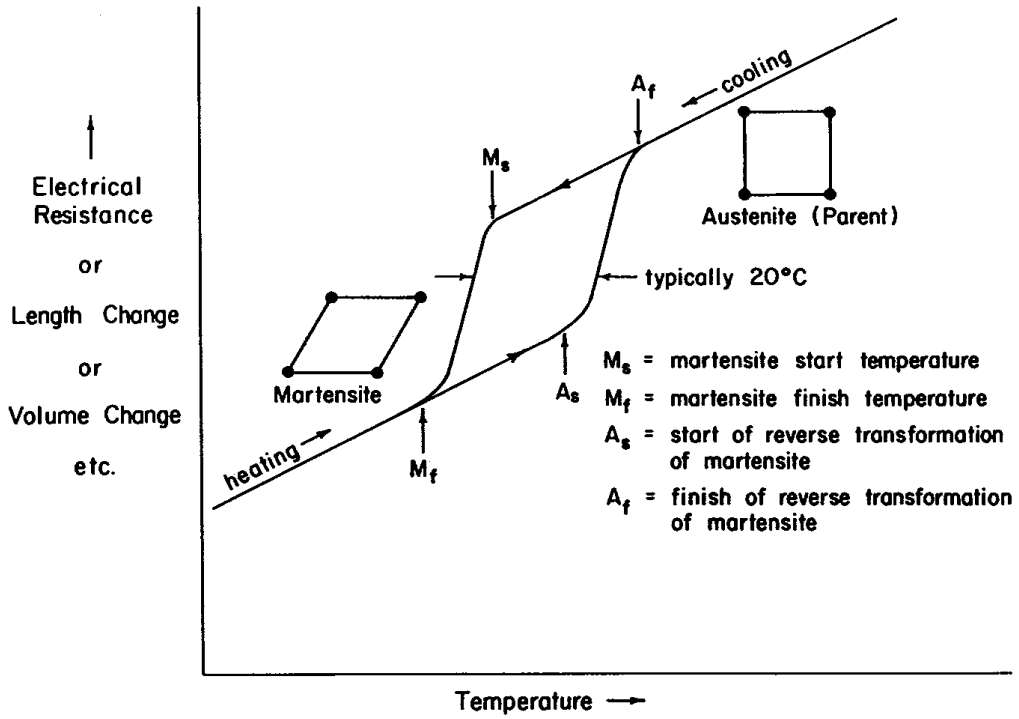


Figure 2-4 Temperature Dependence of Physical Properties [3].

Note that the heating portion of the curve in Fig 2.4 is equally applicable to strain if the initial deformation of the martensite is compressive. For example, given a spring made of SMA, if the spring is compressed in the martensite phase (while cool), then it will expand rapidly at A_s, effectively restoring the "remembered" shape. The curves have a small non-zero slope below A_s and above A_f because like any metal, SMA exhibits normal thermal expansion. For tensile deformations, the curve of Fig 2.4 is inverted, as shown in Fig 2.5. An SMA wire loaded in the low temperature martensite phase will stretch, yielding a large strain; as the wire is heated past A_s, it will contract to restore its original shape.

It is important to note that the hysteresis shown in Fig 2.4 is determined by completely cycling the alloy. In order words, the cooling curve is obtained by cooling the alloy from above A_f to below M_f, and the heating curve is obtained by heating the alloy between these temperatures. The resulting hysteresis is known as the major hysteresis.

If the SMA is only partially cycled i.e. if the temperature switches between heating and cooling within the range M_s to A_s, then minor hysteresis loops result. For example, consider Fig 2.5, which illustrates the strain-temperature characteristics of an SMA wire subjected to a constant tensile load. The solid curve from point 3 to point 1 (following the arrows) and back to point 3 represents the major hysteresis loop, since the wire is heated from a temperature below M_f to a temperature above A_f and then back again. The dotted curve $1 \rightarrow 2 \rightarrow 1$ (following the arrows) is a

minor hysteresis loop arising when the wire is reheated after being only partially cooled. Similarly, the dotted curve 3→ 4 → 3 is a minor hysteresis loop due to the wire being cooled after partial heating.

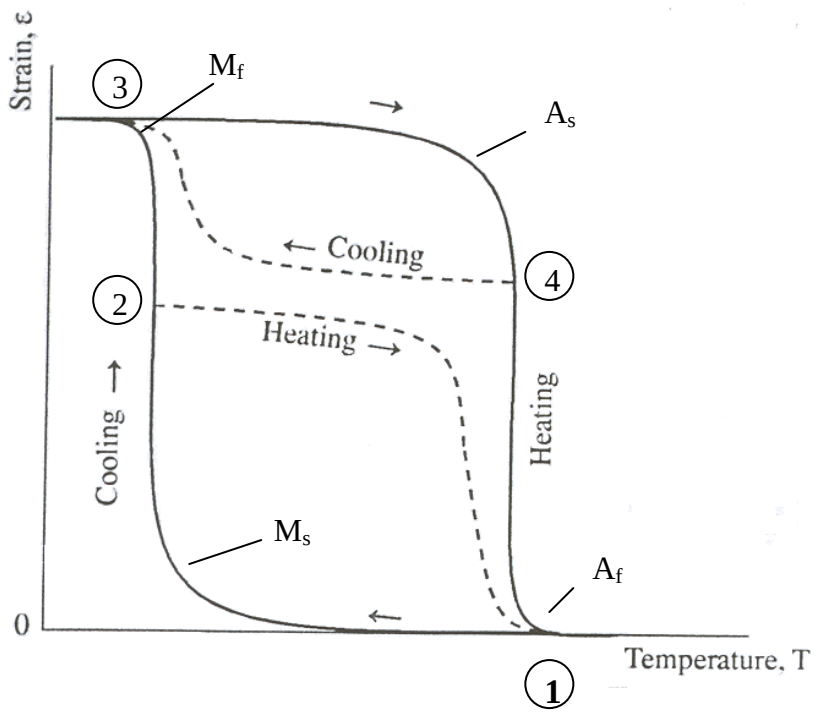


Figure 2-5 SMA Strain-Temperature Relationship [5].

According to Humbeeck et al. [3], continued cycling within the temperature region M_s and A_s will actually shift the hysteresis over a number of cycles until the alloy eventually acquires the shape it normally has in either the fully austenite phase or the fully martensite phase. "As a consequence, regulating mechanisms relying on partial cycling with low amplitude temperature fluctuations (2 to 5°C) are almost impossible unless the material can be regularly transformed completely into one of the two phases, either beta [austenite] or martensite. It can be shown that the device can then be brought back into its original position." [3]

The shifting phenomenon can be explained in the context of minor hysteresis loops. Consider Fig. 2.6. The solid curve indicates the strain-temperature relationship when the wire is partially cycled numerous times. The arrows indicate the direction of motion. The dotted curve illustrates the major hysteresis curve and its only present as a reference. In the figure, the SMA wire is cool initially, such that the strain is at point 1. As the wire is heated, the strain decreases until point 2 is reached. The temperature at this point is less than the austenite finish temperature, A_f, so the wire has not contracted completely. At point 2, heating is discontinued and the wire is cooled. Cooling continues until point 3 is reached. At this point, the temperature is still greater than the martensite

finish temperature, M_f, so the wire has not extended fully. Now the wire is heated until point 4 is reached. The cooling-heating cycle continues for several cycles. The strain-temperature curve progresses to successive points (5,6,7…). From point 4 onward, the temperature is cycled over a similar range. Each time the temperature is cycled, the strain does not necessarily return to the same value it had at the beginning of the cycle. Hence, each minor hysteresis loop is slightly different and the net result is a shift in the strain-temperature characteristic either upward or downward. The only way to return to the major hysteresis loop is to either cool the wire to M_f or heat it to A_f. Note that this scenario is particularly likely when the temperature of the wire is controlled by electrical current – if the current is cycled too quickly, the wire will not be able to cool fast enough and will gradually heat – resulting in a shift in the strain-current relationship toward small strains.

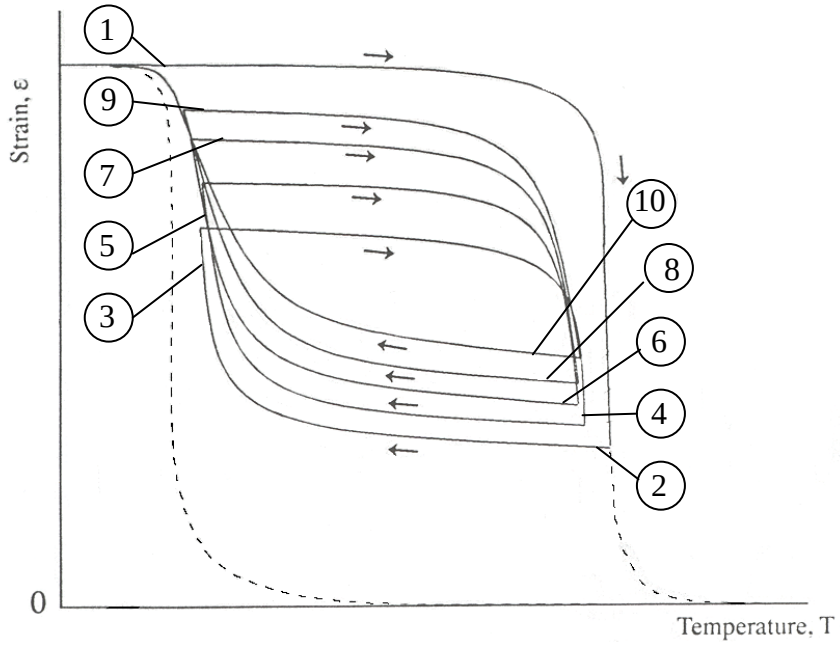


Figure 2-6 Shifting phenomenon and Partial Cycling.

In actuator applications, this shifting phenomenon precludes the possibility of using open-loop control unless longer than required heating and cooling homes are used to ensure the A_f and M_f temperatures are attained, since it is difficult if not impossible to predict the position of the SMA wire. Closed-loop control is necessary for fast cycling. The shifting might be controlled by monitoring the actuator position or force, along with the electrical resistance of the shape memory alloy, in order to determine the instantaneous “operating point” of the alloy and respond accordingly. Alternatively the introduction of an integral term to a controller would also compensate for the shifting effect by progressively increasing (or decreasing) the alloy temperature until the shape memory material responded appropriately.

2.5 Martensite

If the temperature is below M_f, the shape memory alloy will consist entirely of twinned martensite. The twinned martensite, like any metal, has an associated elastic modulus, so the stress-strain curve for low applied stress is approximately linear. If the stress is removed at this point, the material recovers its original shape, like any elastic material stretched within its elastic region. Fig. 2.7 illustrates the stress-strain relationship of martensite. The linear portion of the curve prior to point A corresponds to the elastic region of the twinned martensite.

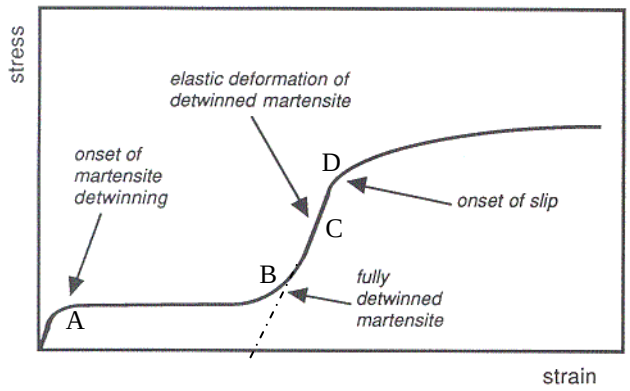


Figure 2-7 Typical Stress-Strain Relationship of Martensite [3].

Since twin boundaries are quite mobile, a larger external stress can be mechanically detwin the martensite. In order words, an externally applied stress can force all variants of the martensite into a single variant. This results in a substantial shape change with a very small increase in stress. Thus, there is a plateau in the stress-strain relationship, as shown in Fig. 2.7 between points A and B. If the external stress is removed then the martensite will recover slightly but remain deformed. This is akin to plastic deformation. However, with the shape memory alloy, heating will cause the original “remembered” shape to be fully recovered.

Once the martensite is fully detwinned (point B in Fig. 2.7), it exhibits elastic behaviour when further stress is applied – the atomic bonds within the rhombic crystalline structure are stretched. This “ elastic” region is marked as C in Fig. 2.8. Thus if the stress is removed, the martensite will return to its unstretched, fully detwinned form, as indicated in the figure by the dotted line.

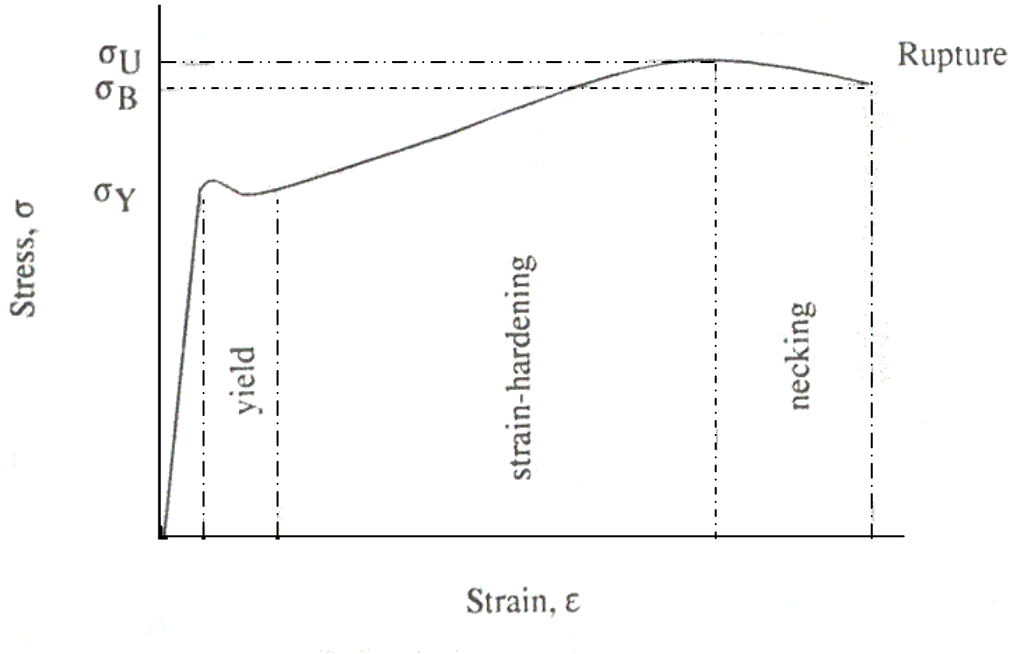


Figure 2-8 Typical Stress-Strain Relationship of Low Carbon Steel [3].

If too much stress is applied, then slip begins to occur – atomic bonds are broken. This process is synonymous with yielding in steel. In this case, permanent irrecoverable deformation has occurred. Point D in Fig. 2.7 marks the point where slip occurs. A typical stress-strain relationship for low carbon steel is shown in Fig. 2.8 to illustrate the difference in behaviour between the martensite of a shape memory alloy and steel.

2.6 Stress-Induced Martensite and Superelasticity

In an unloaded SMA, martensite formation begins when the temperature falls below the martensite start temperature, M_s. It is also possible to induce martensite formation above M_s by the application of sufficient external stress. The applied stress mechanically deforms the austenite crystal lattice into the rhombic structure and may be fully detwinned, so the SMA elongates in the direction of the applied stress. Martensite formation caused by the application of an external stress is known as stress-induced martensite (SIM). The change in shape differentiates stress-induced martensite from thermally-induced martensite, in which no shape change occurs because of twinning. In Fig. 2.2 the process of forcing austenite into a detwinned martensite is the reverse of the normal transformation that occurs upon heating detwinned martensite. Suppose the temperature is further reduced to the martensite start temperature M_s, while the load is still applied. Since the material was forced into the martensite phase by the applied load, no phase change occurs as the temperature lowers-the material is already martensite.

If the applied stress is removed at this lower temperature, no shape change occurs. To recover the original shape, the shape memory alloy must be heated above the austenite start temperature, A_s. Fig. 2.7 illustrates this behaviour with a dotted line. At point B the martensite is fully detwinned due to the applied load. As the stress is reduced, the stress-strain characteristic follows the dotted line and only a negligible change in strain occurs.

If the applied stress is removed while the temperature is above A_s then the martensite becomes unstable, and reverts to austenite. But with the reversion to austenite, the shape changes back to the "remembered" shape. Thus, the material appears to behave elastically because it reverts to its original shape after deformation. The effect is known as superelasticity because the shape change under the applied stress may be as high as 8 %, but the original shape is still fully recovered upon removal of the stress.

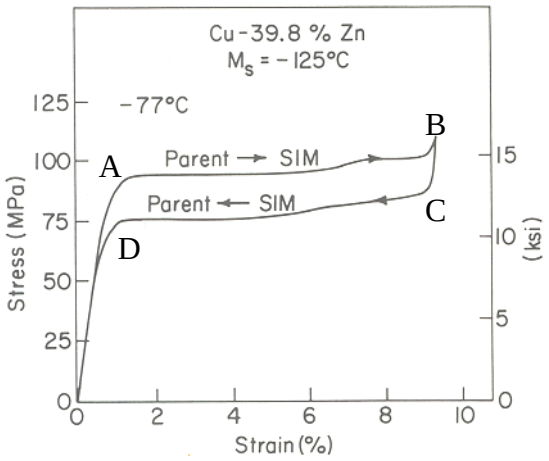


Figure 2-9 Stress-Strain Relationship in a Superelastic Alloy [3].

Fig. 2.9 shows the stress-strain relationship for a superelastic alloy. When no stress is applied, the SMA is in the austenite form since the temperature is above M_s. Thus, if a small stress is applied, the material simply behaves elastically. Once a large enough stress is applied, the formation of martensite is induced. The point at which martensite formation is induced is marked as point A in Fig 2.9. The stress-induced martensite is readily deformed into its fully detwinned states (point B in Fig. 2.9). Very little additional stress is required to cause a significant elongation of the material (approximately 8 % for Ni-Ti alloys). This behaviour is represented by a plateau in the stress-strain curve. Once the martensite is fully detwinned, it begins to behave elastically, resulting in the end of the plateau. It is instructive to compare Fig. 2.7, which illustrates the stress-strain relationship of martensite, to Fig. 2.9, which illustrated the superelastic stress-strain behaviour. If applied stress is now lessened, the elastic curve of the fully detwinned martensite is followed in reversed. Once a critical stress is reached, the martensite becomes unstable (since the temperature is above A_s) and

reverts back to the austenite form. Point C in Fig. 2.9 marks the point at which this instability occurs. Since the austenite form can only exist in the “remembered” shape, a large shape change occurs due to this transformation from stress-induced martensite back to austenite. Again, this results in a plateau in the stress-strain relationship. This plateau occurs at a lower level of applied stress than the austenite-to-martensite plateau. In other words, the material is again exhibiting a hysteresis but in this case it is a mechanical hysteresis rather than a thermal one. Once the martensite has fully reverted fully to the austenite form, the stress-strain curve returns to the elastic curve of the austenite (point D in Fig. 2.9).

The critical stress at which martensite formation is induced is temperature- independent. The higher the temperature, the more difficult it is to force martensite to form by applying an external stress. This temperature dependence is demonstatred in Fig. 2.10, which illustrates the stress-strain characteristics of a Cu-Zn SMA when it is deformed at temperature above its martensite start temperature. For clarity, only the austenite to stress-induced martensite portion of the superelastic hysteresis is shown. Note that while Cu-Zn exhibits much lower transformation temperatures (~ -125 °C) than Ni-Ti (~40-70 °C), qualitatively their stress-strain temperature characteristics are similar.

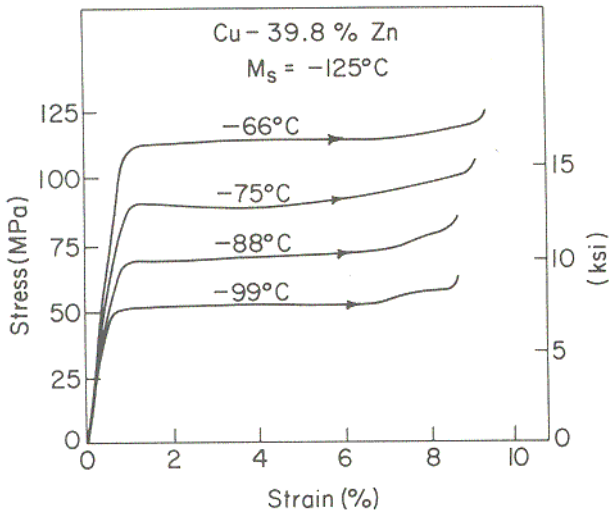


Figure 2-10 Stress-Strain Curve for SMA loaded above Ms[3].

Clearly, any application of shape memory alloys that involve external loads must take into account the formation of stress-induced martensite. Fortunately, a well-defined relationship exits between the temperature and the stress required to induce martensite formation. The relationship is actually linear, the stress required to induce martensite increases linearly with increasing temperature. Or, equally valid, the temperature at which martensite begins to form, M_s, increases with increasing

applied stress. Furthermore, the other transition temperatures obey the same relationship. For example, when an external stress is applied to a shape memory alloy, the temperature at which austenite begins to form, A_s, is higher than in the unloaded alloy. The relationship may thus be generalized to apply to all transformation temperatures. A plot of M_s and A_s versus applied stress is shown below.

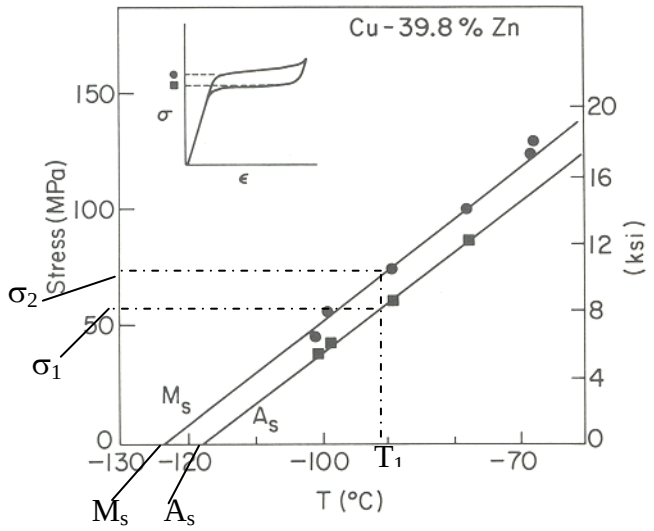


Figure 2-11 Stress dependent of transformation temperature [3].

Another means of describing this linear relationship between stress. σ, and transformation temperatures, T_r, is to state that $d\sigma/dT_r$ is constant. This derivative is known as the stress-rate. Typical values of the stress rate for Ni-ti alloys are 2.5 to 20 MPa/°C [1, pg.28,369-393]. The stress-rate is general property of SMA. The stress rate is related to other properties of the material in the next section.

The superelastic curve is shown in Fig. 2.11. The linear relationship between transition temperatures and applied stress affects the position of the hysteresis loop of the superelastic stress-strain curve. Since the upper plateau is a result of austenite being mechanically deformed into martensite, its level depends directly upon M_s. For example, suppose the material is at temperature T_1 in Fig. 2.11. The stress required to induce martensite from austenite at this temperature is σ_1, because at this stress the martensite start temperature M_s, is equal to the temperature of the wire, T_1. At higher stress levels, M_s, is equal to the temperature of the wire, T_1. Hence, T_1 becomes less than M_s more martensite forms. Similarly in lower plateau, the martensite becoming unstable and austenite beginning to form. Thus, when the wire is at temperature T_1, the stress at which martensite becomes unstable is σ_2 because at this stress level, A_s is equal to T_1. From Fig. 2.11 it is clear that $\sigma_1 < \sigma_2$ i.e., the stress at which martensite returns to austenite is lower than the stress required to induce martensite ~ hence the observed hysteresis.

Note in Fig. 2.11 that the linear relationship terminates at a temperature M_d. This is the temperature above which martensite formation can no longer be induced. The austenitic stress-strain relationship shown in Fig. 2.12 is the result of forcing martensite to form by applying stress to the material.

2.7 Austenite

Austenite is the rigid, approximately body-centered cubic lattice structure assumed by a SMA above A_f. In the austenite form, the SMA exhibits properties very similar to those of steel, or standard piano wire. The austenite behaves elastically up to a certain level of applied stress. When that level exceeded, slip occurs in the austenite crystal lattice and permanent deformation results. The austenite has yielded. The stress-strain relationship for the austenite phase of SMA is show in Fig. 2.12.

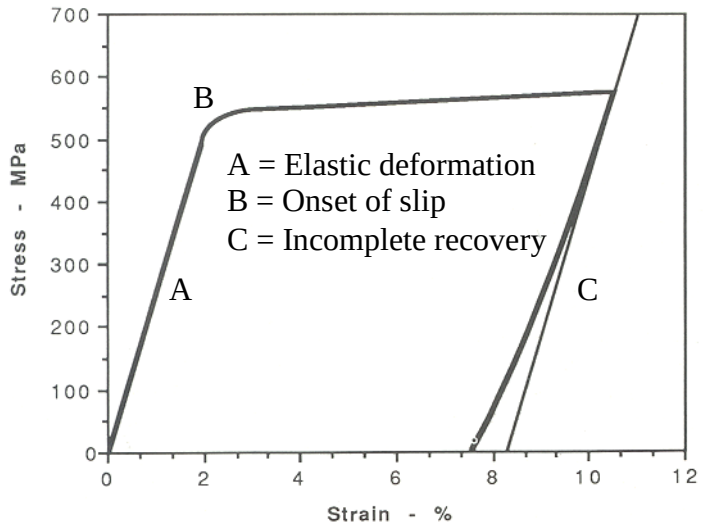


Figure 2-12 Stress-Strain Relationship of Austenite [3].

If the stress is removed after slip has occurred, the austenite will recover slightly but will remain deformed. Unlike martensite which has been deformed to its fully detwinned state, heating will not restore the original shape after slip. If even greater stress is applied the material will crack.

2.8 Two-Way Shape Memory Effect

In one-way shape memory, no shape change occurs during the austenite to martensite transformation that occurs upon cooling below M_s because of twinning of the martensite. However after the SMA has been through a number of cool-deform-heat cycles, microstresses develop in the alloy. These microstresses bias the formation of martensite upon cooling such that only a few variants of martensite are formed, instead of fully-twinned martensite. In order words, the microstresses discourage martensite twinning. Thus, a shape change occurs upon cooling. In essence, the SMA becomes capable of "remembering" two shapes: one formed upon heating, the other formed upon cooling. This effect is appropriately called two-way shape memory.

Since the martensite formed upon cooling is only partially detwinned, the maximum strain recoverable is much less than the recoverable strain in one-way shape memory (at most 5%, typically 2%). Also, the force, or stress, exerted by the shape memory alloy during the transformation to martensite is much less than the force exerted in the one-way shape memory effect (martensite to austenite). Still, the two-way shape memory effect has been used successfully in applications, such as temperature controls for ventilation systems, air conditioners, and reversible fasteners [3].

The microstresses that cause the two-way shape memory effect can be produced by special treatment or training of the shape memory alloy material. There are several techniques for training a shape memory alloy to exhibit two-way shape memory:

1. Repeat cool-deform-heat cycles until the alloy begins to exhibit the two-way shape memory effect.
2. Over-deform the alloy in the martensite form. Upon cooling, the alloy will partially recover the overdeformed shape.
3. Deform the alloy in the martensite form and constrain it. Then heat above A_f and cool it below M_f repeatedly while it remains constrained in the deformed shape.

Other variations exit, but the last method given proves to be fairly effective. The first method is interesting because it means that any SMA used as an actuator is likely to exhibit the two-way shape memory effect after repeated used. This may slightly improve the efficiency of such an actuator.

2.9 R-Phase

The two principal phases of SMA, the parent phase (austenite) and the product phase (martensite) have been examined. A third phase, known as the R-phase, is also present at temperatures above M_s. The R-phase is manifested as an elongation of the cubic austenite lattice into a rhombohedral form (hence the name R-phase). The deformation is shown graphically in Fig. 2.13. The angle α, is called the cube angle and varies from 90° down to approximately 89.3°.

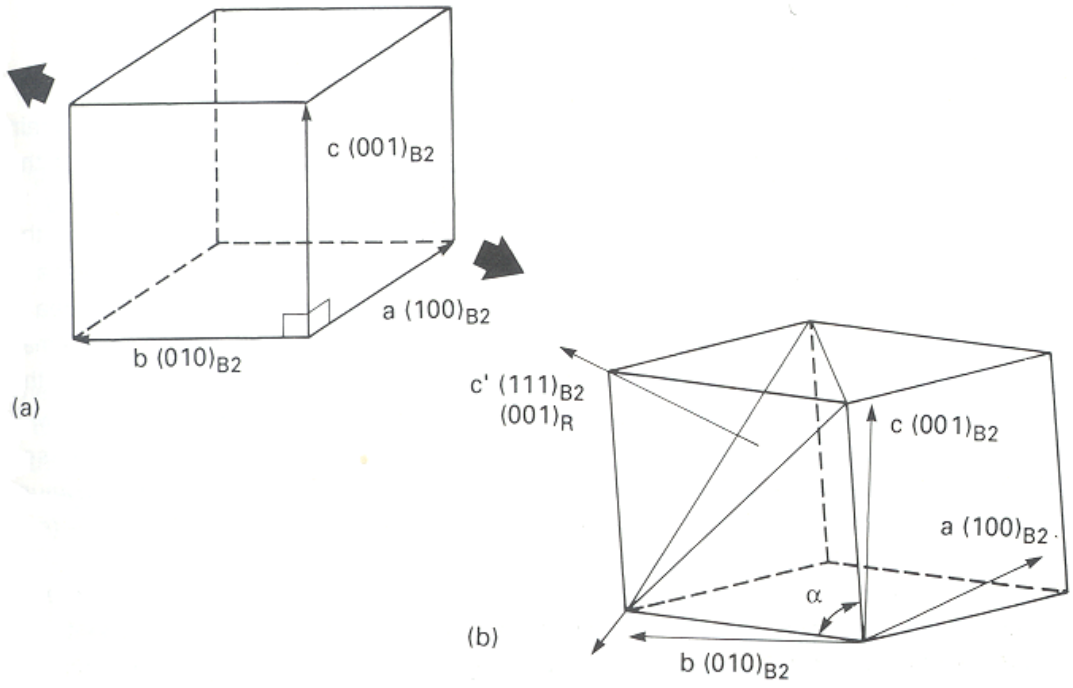


Figure 2-13 R-Phase Transition – Microscopic Phenomenon [3].

The R-phase transformation, like the martensitic transformation, is a thermal effect. The R-phase begins to form from austenite at the R-phase start temperature, R_s, somewhat above A_f. As the temperature decreases, the deformation of the lattice increases. Transition to the R-phase is complete at the R-phase finish temperature, R_f.

Upon heating, the crystal lattice returns to the austenite form but the temperature at which austenite begins to form is slightly higher then R_s. In order words, just like in the martensitic transformation there is a hysteresis. This hystersis is shown in Fig. 2.14.

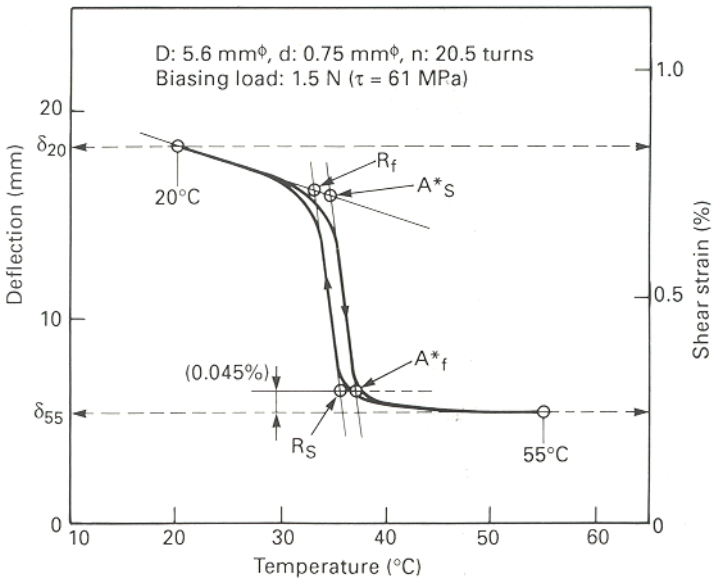


Figure 2-14 Strain verse temperature for the R-Phase while under load [3].

In this figure, the SMA is tested under a constant of 1.5N. Thus, the alloy elongates under the applied load when the temperature is decreased and the R-phase is formed, and the alloy contracts to the "remembered" shape when the temperature is increased and austenite forms.

The temperatures A_s' and A_f' refer to the temperatures at which austenite begins to form from the R-phase and at which the SMA has become entirely austenite respectively. These are not the same temperatures as the A_s and A_f temperatures of the martensitic transformation.

The hysteresis is only about 1.6 °C wide - a much smaller hysteresis than the hysteresis in the martensitic transformation (more than 10°C). Clearly, the smaller hysteresis would improve the efficiency and speed of operation of an actuator. Unfortunately, the maximum recoverable strain in the R-phase is only about 1%-much smaller than the recoverable strain in the martensitic transformation.

Since the R-phase is negligible in SMA designed to exhibit a large one-way shape memory effect, and since larger strains are generally desirable in robotic applications, further investigation of the R-phase is left to the reader.

2.10 Fatigue Life

Any long term application of a SMA must account for the fatigue life of the material i.e., the number of loading-unloading cycles before the material fails. From the preceding discussion it is

clear that the fatigue life will depend upon the load applied. This leads to a stress versus number of cycles (SN) curve for the fatigue life of the alloy. A typical fatigue characteristics for steel is shown in Fig. 2.15.

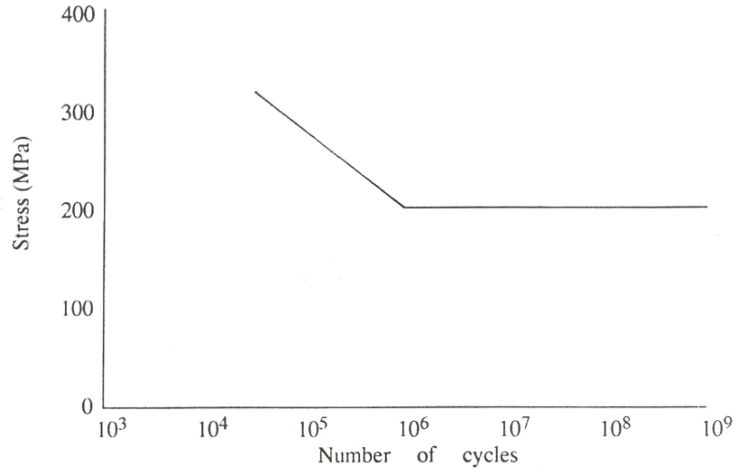


Figure 2-15 Typical Fatigue Characteristics of Steel [3].

For a SMA, the fatigue life also depends on the mode of operation. In other words, either the alloy is being thermally cycled (the material is repeatedly heated and cooled in order to produce the one-way shape memory effect) or mechanically cycled (the material is held at a constant temperature and loaded and unloaded) affects the fatigue characteristics. If the alloy is mechanically cycled then the temperature at which the material is cycled also changes the fatigue characteristics because the temperature determines the phase of the material. The fatigue properties of martensite, stress-induced martensite, the R-phase, and austenite all differ.

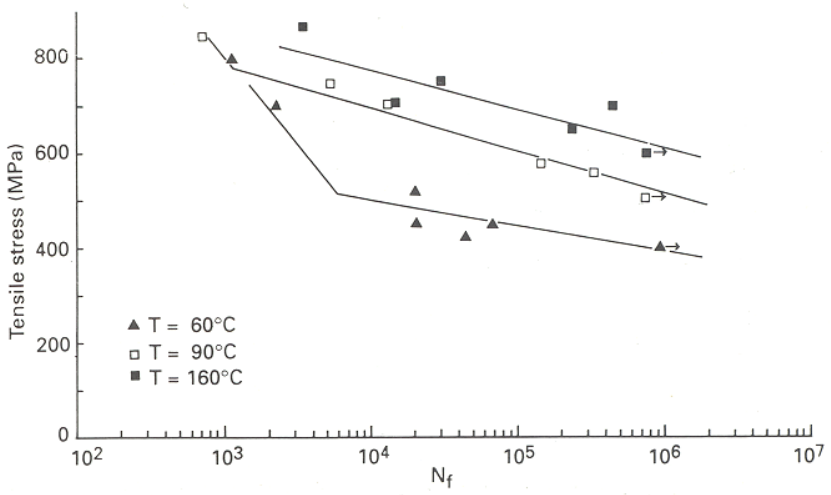


Figure 2-16 Fatigue Life of Ni-Ti (50.8% Ni, 400°C anneal for 1 hr) [3].

Fig. 2.16 shows the SN curves for a Ni-Ti alloy which is mechanically cycled at three different temperatures [3]. The processing treatment of the alloy is also mentioned in the figure because the fabrication process affects the fatigue life. Note that the martensite (T = 60°C) behaves similarly to steel. One might have concluded that normal steel generally exhibits a longer fatigue life. However, one must take into account the different stress levels. At a stress of approximately 207MPa, the steel reaches its endurance limit – the stress below which the steel may be cycled indefinitely. At 276MPa, the steel will failure after approximately 10^5 cycles. For this same value of stress (276MPa), the Ni-Ti alloy does not rupture after 10^7 cycles at any operating temperature. The Ni-Ti alloy only ruptures after 10^5 cycles if the applied stress is over 450MPa and it is mechanically cycled in the martensite phase. According to McNichols et al. [3], the fatigue life of the Ni-Ti alloy when it is thermally-cycled is even higher – more than 10^7 cycles.

The fatigue lifetimes of a Ni-Ti SMA are clearly sufficient for most actuator applications. However, there are other factors which contribute to failure of a SMA actuator design before failure of the alloy itself occurs. For example, thermal and mechanical cycling can also cause deterioration of the recovery stress – the stress exerted by the alloy when it is heated. If an actuator depends on a certain minium recovery stress being maintained throughout its “lifetime” then the designer must be aware of the deterioration of the recovery stress and other properties due to thermal or mechanical cycling. This observation led Suzuki and Tamura[3] to the following definition: the lifetime of a shape memory alloy actuator is the number of deformation cycles until the recovery force of the shape memory actuator is 70% of its initial value. Using this definition, the lifetime of a Ni-Ti alloy operating exclusively in the R-phase region (no martensite phase during cycling) is still over one million thermal cycles. Since Suzuki and Tamura concentrate on increasing the lifetime of a SMA actuator by restricting its operation to the R-phase, the lifetime of a SMA actuator employing the martensite transformation is not discussed here. It is however, dependent upon the applied load as might be expected. The R-phase fatigue is virtually independent of load because the maximum strain possible in the R-phase is only about 1%.

2.11 Corrosion behaviour

In the galvanic series, Ni-Ti based alloys as a family are slightly more noble than 316 stainless steel, and show similar corrosion behaviour. The excellent corrosion resistance is provided by a naturally formed thin adherent oxide layer known as a passive film. This film is very stable, so the Ni-Ti alloys are resistance to many forms of corrosive attack. However a warning should be made at this point, the breakdown of this passive film can occur, if there is a contact with some aggressive solutions such as highly acidified chloride solutions. Nevertheless if Ni-Ti is to be used

in these conditions, some form of protective coating is advisable. However this is not of much concern, as the research with this nitinol wire mechanism doesn't involve a corrosive environment.

2.12 Limitation

The wires' prime function is to contract in length and create force or motion when heated. There are limits, of course, to how much force or contraction can be obtained. The shape-memory transformation has a natural limit in the NiTi system of about 8 percent [1]. That is the amount of strain that can occur in the martensite phase by the reversible martensitic twinning that yields the memory effect. Deformation beyond this level causes dislocation movement throughout the structure, and then that deformation is not only irreversible but degrades the memory recovery as well. For materials expected to repeat the memory strain for many cycles, it is best to use a cyclic memory strain of no more than 4 to 5 percent.

The force that the wire can exert when heated is limited by the strength of the austensitic phase. The phase transformation, or crystal change, that causes the memory effect has more driving force than the strength of the parent material, so one must use care not to exceed that yield strength. The yield strength of the wire in austensitic phase is more than 344.7 MPa (50,000 psi), and on a single pull the wire can exert this force. To have repeated cycling, however, one should use no more than two-thirds of this level, and forces of 137.9 MPa (20,000psi) or less give the best repeat cycling with minimal permanent deformation of the wire.

2.13 Temperature gradient

The temperature distribution existing within a material can at most depend on three space variables and on time. If the temperature is not a function of time, the problem is referred to as steady or the temperature distribution as being in a steady state. If indeed the temperature is a function of time, the problem is called unsteady and the temperature distribution is referred to as being transient. Typically, internal heat generation within a solid can be from chemical reactions, as the result of an electric current passing through the material, or by a nuclear reaction. Some of the energy passing through the control volume may be stored, thus increasing the internal energy of the material, which is sensed physically as an increase in the temperature of the material. Thus nitinol wire is a typical case for unsteady heat flow.

During in transient conduction the temperature distribution is not uniform until the steady state temperature distribution is reached within a system. The unsteady temperature distribution is named the temperature gradient. Because of the existence of a temperature gradient within a

system, internal strains are generated. The size of gradients in the wire (cooled only by convection) is determined by a dimensionless number known as the Biot number. This is given by

$$Bi = \frac{hL}{k} \tag{2.1}$$

The Biot number is a dimensionless ratio of convection to conduction resistance to heat transfer. Thus the Biot number gives an indication of the temperature drop within the solid compared to the temperature difference between the solid surface and the fluid. Where h is the heat transfer coefficient governing the convective cooling and L is the characteristic dimension of the object being cooled. For thin samples $2L$ is the sample thickness, while for a long cylinders L is the cylinder radius.

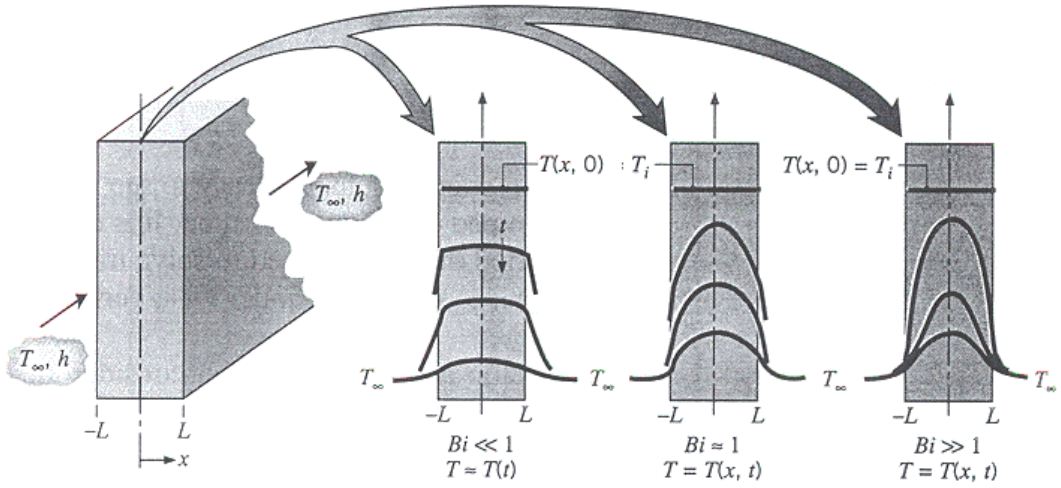


Figure 2-17 Relationship between the Biot number and the temperature profile [6].

The influence of the Biot number on the transient temperature distribution is shown in Fig. 2.17. Shown are three case of interest; consider the plane wall, which is initially at a uniform temperature T_i and experiences convection cooling when it is immersed in a fluid of $T_\infty < T_i$. When Biot number is less than 0.1, virtually all the temperature difference is between the solid and the fluid, and the solid temperature remains nearly uniform as it decreases to T_∞. For moderate to large values of the Biot number, however, the temperature gradients within the solid are significant. Hence T = T(x, t). Note that for Bi >> 1, the temperature difference across the solid is now much larger than that between the surface and the fluid. Hence the temperature gradient is now significant and no longer could assume to be uniformly distributed.

Applying eqn 2.1

$$Bi = \frac{hL}{k} = \frac{75 \times 7.62 \times 10^{-5}}{8.66} = 6.6 \times 10^{-4}$$

According the calculation from above, the Biot number for the nitinol wire is far less than 0.1, in this case determination of the temperature gradient is unnecessary. Thus the lumped capacitance method is appropriate for the determination of this transient conduction problem if necessary, and the next section examines modelling using the lumped capacitance method. However an initial investigation of the exact modelling (See Appendix F) is beyond the scope of this project, and is suggested as a topic for further research.

2.13.1 Temperature –Current relationship

The temperature–current relationship is derived below using several simplifying assumptions. The most important assumption is that the heat capacity may be considered a constant lumped parameter. As the crystal structure of a SMA wire changes from martensite to austenite, or vice-versa, the specific heat capacity does not remain constant. Furthermore, heat capacity is physically a distributed parameter. However, the use of the constant lumped parameter is assumed in order to significantly simplify the analysis. Nevertheless as it has been pointed out in the previous section the Biot number for the nitinol wire is far less than 0.1, so in this case determination of the temperature gradient is unnecessary.

The temperature-current relationship is thus the direct application of conventional dynamic conduction and heat transfer models. The change in internal heat energy of the SMA wire must equal the energy delivered to the wire by the electrical current minus the energy transferred to the air by natural convection. Thus, in mathematical notation, the energy equation may be expressed as:

$$\rho c V \frac{dT(t)}{dt} = Ri^2(t) - hA\left(T(t) - T_\infty\right)$$

Temperature (T) and current (i) are shown explicitly as functions of time (t) to emphasize that the other parameters are considered constants. Note that the volume (V), density (ρ), heat capacitance (c) and surface area of the wire will change during the SMA actuator operation but it is assumed that these effects are negligible.

2.14 Thermal effects

External loads are not the only sources of stresses and strains in the nitinol wire. The effect of a change of temperature on the nitinol wire is a small change in size and hence strain. This can, in some circumstances, induce considerable internal stress. These stresses lead to degrade of the material strength. Thus an investigation of these stresses was carried out.

The phenomenon is that the contraction of the nitinol wire is due only to heating and the relaxation only to cooling. Both contraction and relaxation are virtually instantaneous with the temperature change of the wire. As a result, mechanical cycle speed is dependent on and directly related to temperature changes. Applying high currents for short periods of time can be quickly heat the wire. However in a finite time interval, heat that was generated at the centre propagates towards the outer surface. In a finite difference of radius, the difference of temperature would cause an internal shear stress upon $d\phi$. This is due to the fact that the heated material up to the radius r would undergo a thermal expansion (though it contracts along z axis, but it expands radialy), whereby the elements from r + dr to r_0 were cooled by the ambient. Along with characteristic of shape memory effect causes the wire to contract thus an axial tensile stress would induced. If the wire is made to contract too fast with a load, the inertia of the load can overstress to the wire. These effects also take place in an opposite sense upon cooling. Consequently fatigue is the resulting side effect. Fatigue that is damaging to the nitinol wire would usually show up on the form of wire elongation or reduced stroke within the first few hundred strokes. Thus the best rule of thumb is to use enough nitinol wire to be sure one is well within the parameters in which it can work.

A brief theoretical study of the transient conduction was carried out and is contained in the Appendix F. With the calculated temperatures, the coefficient of thermal expansion and the assumption that the wire behaves axis symmetrically, thermal strain could then be calculated. (Notice that care should be taken when calculating the thermal strain, as the coefficient of thermal expansion are different for both martensite and austenite phases.) From then simply apply Hooke's law to obtain the stresses for each axis.

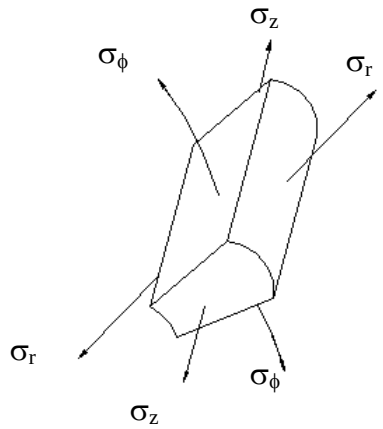


Figure 2-18 Three-dimensional stresses for cylindrical system.

2.15 Conclusion

The detailed technical characteristics and limitation of the SMA has been described in engineering terms. In short, both austenite and martensite phases dominant the shape memory effect. With sufficient heat energy supplied, thermomechanical energy can be produced. If the reversible thermomechanical energy is well utilised along with an appropriate design of the mechanism, it can be just as useful as conventional actuators. The unique shape memory phenomenon has posed an interesting topic for robotic researchers.

The transient conduction of the SMA wire was studied, and was suggested that the lumped capacitance method be used to estimate the differential temperature between the ambient and the SMA wire. This suggestion was valid, as the Biot number was found to be less than 0.1. Hence, when confronted with such a problem, the very first thing done is to calculate the Biot number. If this is<<1, then, error associated with using the lumped capacitance method is small.

In Appendix F an initial investigation for the exact analytical solutions to transient conduction problem was obtained. Although this exact modelling has not been utilised, the result is presented as a starting point for further investigation on the stress induced by the thermal gradient.

Chapter 3 Design of Nitinol Actuated Stewart Platform

3.1 Prototype design

In recent years there has been considerable interest in the application of parallel robots because of their greater stiffness and accuracy when compared to the standard serial robots. However these desirable characteristics are obtained at the expense of a reduced workspace.

The 6 dof (degree of freedom) parallel mechanism known as a Stewart Platform [7] is often attributed to Stewart [8] for use as a motion (flight) simulation platform, but it was originally developed by Gough [9] as a tyre testing machine for the Dunlop Tyre Company. It was later used as a precision manipulator [10] , as a milling machine [11] , and as a singularity free antenna aiming mechanism [12]. A schematic of the basic Stewart Platform is shown in Fig.3.1.

Figure 3-1 A schematic view of the basic Stewart Platform [8].

The prismatic actuators used for the Stewart Platform are usually continuous hydraulic or electrical lead screw actuators. Recent work has investigated the use of simple on off pneumatic actuators to reduce the cost and complexity of Stewart Platform systems.

An even simpler electrically operated actuator based on Nitinol wires has been developed to activate small Stewart Platform. To use the Nitinol wires as a prismatic actuator would requires a constrained preloaded spring with the Nitinol wires up the middle. Termination of the spring constraints with ball joints and electrical connectors for the Nitinol wires results in a somewhat complicated structure, especially if many actuators are required. For example, an eight stage stacked Stewart Platform design would require 48 actuators yielding the possibility of 256E12 unique positions. Construction of such actuators is expensive so another approach was taken.

The approach taken was to design a system that was suitable for rapid prototyping. The Nitinol wire was pre-tensioned using a bow that has ball joints at each end connect to the two triangular platforms of a stage (for a single stage Stewart Platform one is the triangular base and the other is the end effector). This arrangement is shown in Fig. 3.5 where the triangle corners have been

shaped to accept the ball joints. Note that the cups that hold the ball joints into the spherical cavities in the triangular stages were grown in situ during the rapid prototyping process. The bow shape leaf spring were grown with a flat orientation during prototyping in order to develop full strength for flexing as it provides the forces needed to support the platform as well as pre-tensioning the Nitinol wire which acts as a “bowstring”. A detailed description of the bow is given in section 3.7 and subsequent sections. A brief introduction of rapid prototyping is now presented.

3.2 Rapid prototyping

Rapid prototyping is the name given to a host of related technologies that are used to fabricate physical objects directly from CAD data sources. These methods are unique in that they add and bond materials in layers to form objects. Such systems are also known by the general names “solid freeform fabrication” and “layered manufacturing”, and offer advantages in many applications when compared to classical subtractive fabrication methods such as milling or turning:

1. Objects can be formed with any geometric complexity or intricacy without the need for elaborate machine set up or final assembly;
2. Objects can be made from multiple materials, or as composites, and materials can even be varied in a controlled fashion at any location within an object;
3. Solid freeform fabrication systems reduce the construction of complex objects to a manageable, straightforward, and relatively fast process.

These properties have resulted in their wide use to reduce time to market in manufacturing. Today's systems are commonly used by engineers to better understand and communicate their product designs as well as producing tooling to manufacture those products. It is not limited to manufacturing engineer and it can be fitted to other areas such as Surgeons, architects, artists and individuals from many other disciplines

The names of specific processes themselves are also often used as synonyms for the entire field of rapid prototyping. Among these are stereolithography (SLA for stereolithography apparatus), selective laser sintering (SLS), fused deposition modeling (FDM), laminated object manufacturing (LOM), inkjet-based systems and three dimensional printing (3DP). Each of these technologies - and the many other rapid prototyping processes - has their refective strengths and weaknesses. However FDM is the only method of rapid prototyping currently available in New Zealand, therefore attention is focused on this technology.

3.2.1 Fused Deposition Model

FDM is the second most widely used rapid prototyping technology, after stereolithography. A plastic filament is unwound from a coil and supplies material to an extrusion nozzle. The nozzle is heated to melt the plastic and has a mechanism which allows the flow of the melted plastic to be turned on and off. The nozzle is mounted to a mechanical stage which can be moved in both the horizontal and vertical directions.

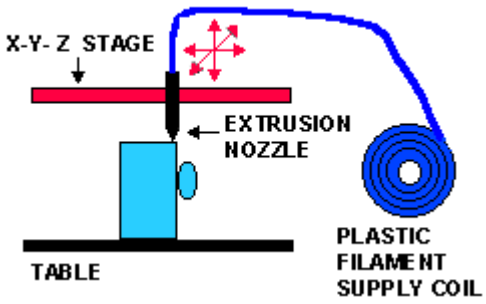


Figure 3-2 Simplify model of Fused Deposition Model.

As the nozzle is moved over the table in the required path, it deposits a thin bead of extruded plastic which form each layer. The plastic hardens immediately after being squirted from the nozzle and bonds to the layer immediately below. The entire system is contained within a chamber which is held at a temperature just below the melting point of the plastic.

Several materials are used with. This process including Acrylonitrile Butadiene Styrene **(**ABS) and investment casting wax. ABS offers good strength, and more recently polycarbonate and polysulfone materials have been introduced to further extend the strength and temperature range of the resultant products. Support structures are fabricated for overhanging geometries and are later removed by breaking them away from the object. A water-soluble support material that can simply be washed away is also available.

The method is office-friendly and quiet. FDM is fast for small parts of the order of 15 ml, or those that have tall, thin form-factors. It can be very slow for parts with large cross sections, however. The finish of parts produced with the method have been greatly improved over the years, but remain inferior to stereolithography. The closest competitor to the FDM process is probably three dimensional printing. However, FDM offers greater strength and a wider range of materials than the implementations of 3DP, which are most closely comparable with the FDM available in N.Z.

The unavoidable draw back of rapid prototyping is that it builds an object in layers, so there is inevitably a "stairstepping" effect produced because of the finite thickness layers. The methods

that produce the thinnest layers have the least stairstepping, but it's almost always visible. Polishing is required in order to achieve a better surface smoothness.

3.3 Design of the Stewart Platform

The main objective for the design of the robot was to construct a robot that would prove the concept of using nitinol wires as binary actuators. Force/stress at the design stage was based on estimates of the stresses experienced by the leaf springs and socket joints and a significant safety margin was included in the design. The robot was essentially designed for rigidity and rapid prototyping, and was hence bulky. This procedure led to over-design with excess weight that could subsequently be reduced to achieve an optimal solution. Engineering drawings of all components discussed in this chapter can be found in Appendix A.

The Stewart Platform design was an attempt to optimise the design parameters while fulfiling the design specifications. The final design ended up as a complex shape, that was modular in design and will suited rapid prototyping as this was the easiest and most cost effective way to produce such detailed component design.

Basically the structure consists of eight prismatic equilateral triangular boxes, two base covers, 24 leaf springs and 48 ball joints and cones etc. (See Fig. 3.3 for a single stage Stewart Platform and shape memory alloy actuators) Each of these main part will be discussed.

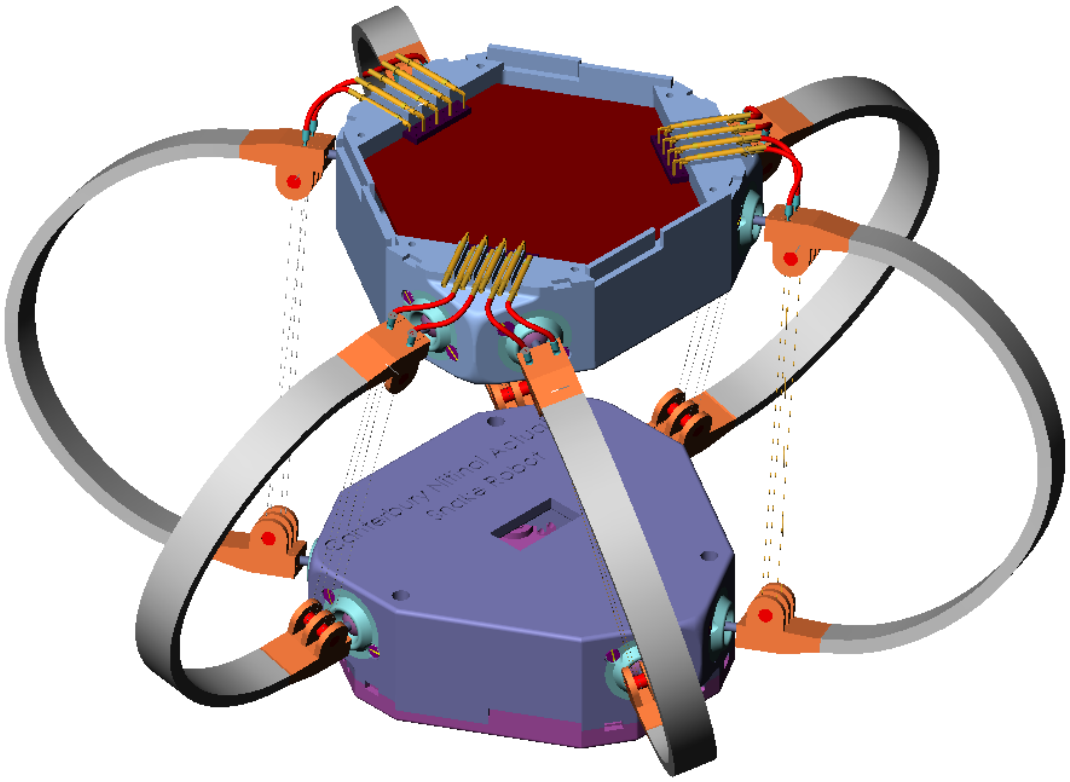


Figure 3-3 A single stage of Stewart Platform.

The platform section was constructed from two prismatic equilateral triangular boxes, each with an open side. These sections were joined together (with the open sides placed face to face) by three M2 cap screws. This formed a closed equilateral triangular box. A single prismatic equilateral triangular box is shown in Fig 3.4

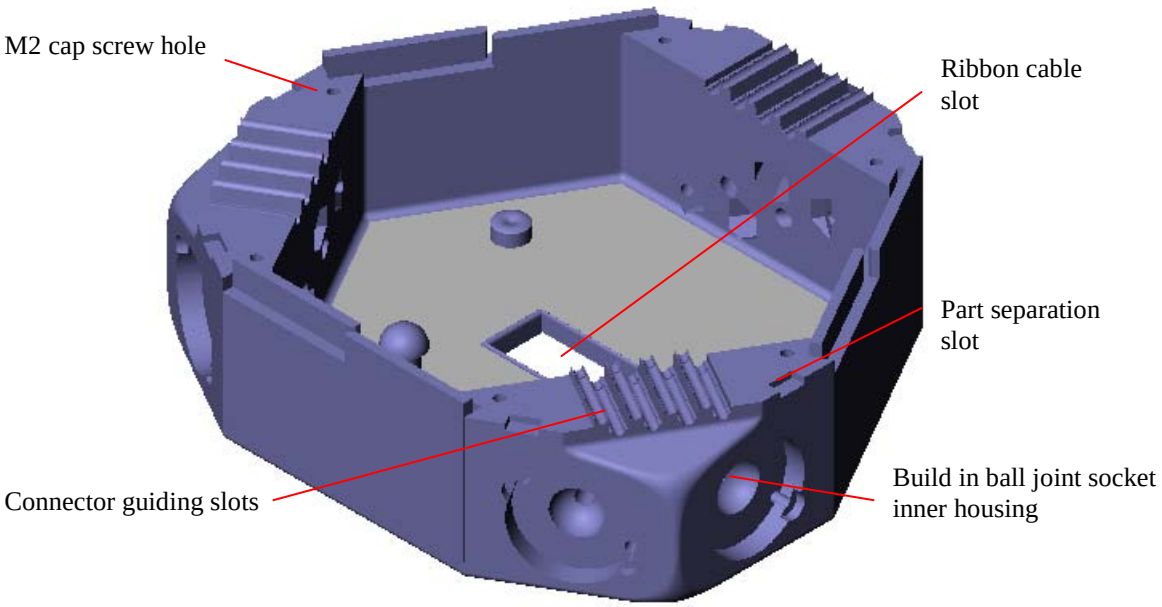


Figure 3-4 A single prismatic equilateral triangular box.

The junction of two equilateral triangular boxes which capsulated an internal space for holding and protecting the circuit board. The length of the prismatic triangular box is 85.03 mm, and the height, when the box closed, is 46 mm (when assembled). Guiding slots at each corner of the box were used for connection of the nitinol wire to the circuit board and also provide support for the connection pins. For dismantling the platform section, two slots were designed at each edge of the box with the dimensions of 4 mm x 2 mm x 2 mm. These slots are used to open the box using a screwdriver. Communications and power supplied to the circuit boards through a ribbon slot. The ribbon slot is located at the bottom center of each triangular box and has a dimension of 9 mm x 18 mm. This arrangement gives space between the nitinol wires and the ribbon cable preventing contact. The Stewart Platform was designed to be modular and as small as practicable with the size being dependent on both the circuit board and the ball joints.

3.4 Design of the base cover

The base cover was designed so that it can be used as either a top platform cover or as a base for necessary locating the manipulators. The base cover retains most of the features in prismatic equilateral triangular box. Since ball joints are not require in the cover, the height of the cover was reduced to 5mm. To ensure the whole manipulators can be located firmly, thicken sections were

grown (using Rapid Prototyping) for the location bolts. Additional reinforcement bars were also grown to strengthen these sections. Fig. 3.5 contains a schematic view of the base cover.

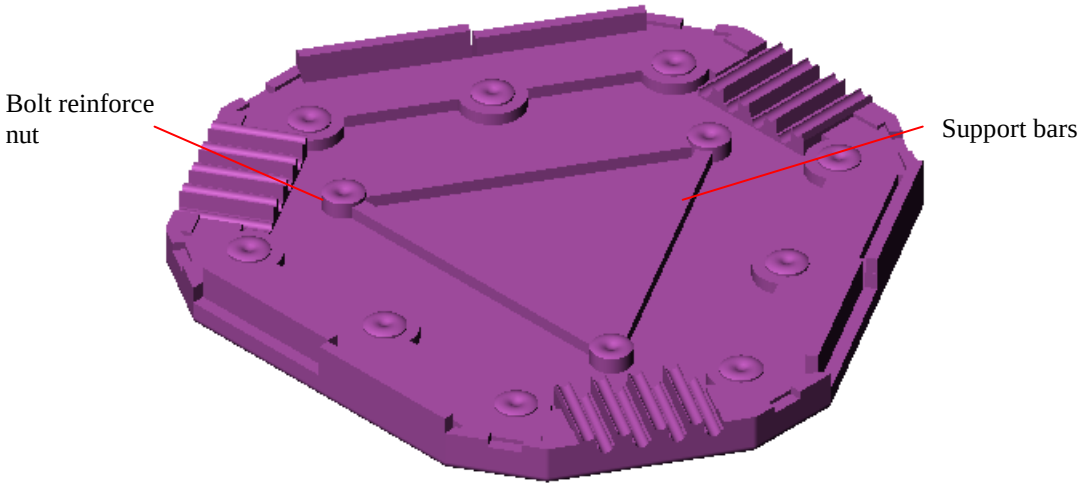


Figure 3-5 A base cover.

3.5 Optimisation of the nitinol wire

The main problem with using Nitinol wire actuators was the cooling time required for the wires to reach ambient temperature. As seen in Fig. 2.4, cooling from the high temperature state to just above the hysteresis knee (at around 40°C) occurs approximately in the constant strain region. If the wire is held at constant strain (at~ 40°C) the wire can cool to ambient temperature in minimum time.

The approach taken was to increase the number of wires while keeping the same total cross sectional area. This remains the same force capability but increases the surface area. A simple proof is shown below. The parameter with the subscript 1 denotes the arrangement of a single nitinol wire with a larger diameter, whereas subscript 2 denotes the arrangement of four thinner nitinol wires.

Given

ρ = resistivity of the nitinol wire [Ωm]

R = electrical resistance [Ω]

A = cross sectional area [m^2]

L = length of the nitinol wire [m]

q = heat transfer rate [W]

V = voltage [volt]

P = power [W]

I = current [A]

T_∞ = ambient temperature [°C]

T_s = surface temperature [°C]

The relationships of electrical resistance R_1 and the relationship of the heat transfer rate of q, the following can be established, such that

$$R_1 = \rho \frac{Length}{Area} = \rho \frac{L}{\pi r^2} \tag{3.2}$$

$$q_1 = h(2\pi r L)(T_s - T_\infty) \tag{3.3}$$

$$q_1 = P_1 = \frac{V_1^2}{R_1} \tag{3.4}$$

$$I_1 = \frac{V_1}{R_1}$$

Assumed diameter of the second arrangement of the NiTi wires is half that of the first and also the length is four times that of the first arrangement, i.e. 4 strands at length L for the actuators, but connected in series rather than parallel.

$$R_2 = \frac{\rho(4L)}{\pi\left(\frac{r}{2}\right)^2} = \frac{16\rho L}{\pi r^2} = 16R_1$$

$$q_2 = h\left(2\pi \frac{r}{2} 4L\right)(T_s - T_\infty) = 2q_1$$

$$q_2 = \frac{V_2^2}{R_2} = \frac{V_2^2}{16R_1} = 2q_1$$

$$V_2^2 = 32 q_1 R_1 = 32 \frac{V_1^2}{R_1} R_1$$

$$V_2 = \sqrt{32} V_1 \approx 5.6\ V_1$$

$$I_2 = \frac{V_2}{R_2} = \frac{\sqrt{32}V_1}{16R_1} = \frac{4\sqrt{2}V_1}{16R_1} = \frac{\sqrt{2}}{4} I_1 \approx 0.354\ \mathrm{I}_1$$

Base on the above calculations, using 4 Nitinol wires of 75μm diameter instead of one of 150μm retains the strength but double the cooling surface, and halving the cooling time. Placing the 4 lengths in series resulted in a 16 times increase in resistance, and the current requirement is reduced. Essentially, the power requirement is doubled to maintain the same temperature, but since the heat capacity is unchanged, the transients power requirements to establish the strain remain unchanged. Maintaining the hysteresis knee temperature requires double the power, but since the system is only 20°C above ambient by using pulse width modulation (PWM), the actual power is still reduced compared to the standard method of maintaining the wire at the maximum temperature around 80 °C or 60°C above ambient. While the power requirement may be doubled, the PWM technique allows the power to be reduced by three times, and thus the power requirement is only 2/3 the normal power requirement, and the switch off times are approximately halved.

3.6 Design of the leaf spring

The function of the leaf springs in this mechanism were not only to provide support between the decks, but also to pretension the nitinol wires. The required assembled shape would be in a shape of a half circle with a diameter of 90 mm. Base on these criteria, the pre-assemble shape of the leaf spring should be in a form of an arc. Once the nitinol wires are wound between both ends of the leaf, the recovery force will straighten up the nitinol wires. However, the opposing force produced by the leaf spring is perfectly balanced by the tensional forces of the nitinol wires. Subject to the accurately tuned tension forces, the nitinol wires should hold the leaf spring in a half circle shape. Thus the relationship between the amount of tension force and the corresponding deflection can be derived.

Consider the simplified case consisting of a curved bar AB has a centerline in the form of a quarter circle of radius R, as shown in Fig. 3.6. From symmetry the bar can be treated as there is a fixed support at A and carries a vertical load P at the free end B.

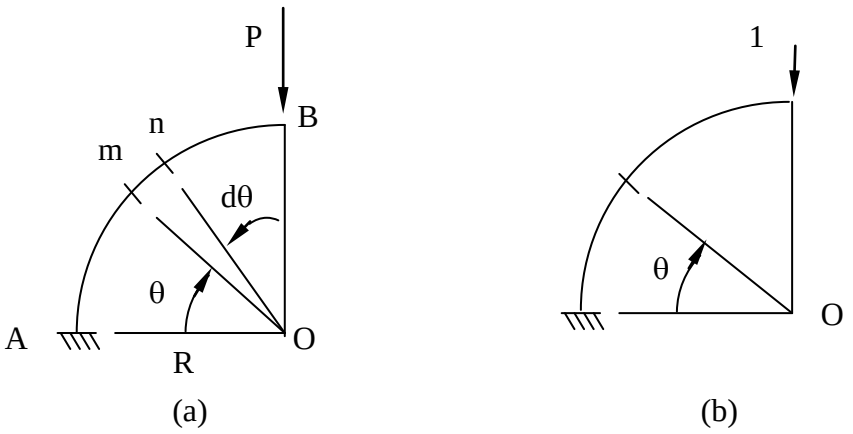


Figure 3-6 Deflection of a curved Bar.

The unit-load equation for finding deflections of the curved bar can be expressed in the general form

$$\Delta = \int \frac{M_U M_L ds}{EI} \tag{3.5}$$

where ds, equals R dθ, and is the length of an element mn of the bar. It is assumed that the bar is thin compared to the radius R so that the above formula, originally derived for the bending of a straight bar, can be used. Also, eqn 3.5 only answers the effects of flexural deformation.

The bending moment M_L caused by the load P is M_L = -PRcosθ, where a positive moment is assumed to caused compression on the outside of the curved bar. Also the bending moment M_U due to a vertical unit load (Fig 3.8 b) is M_U = -Rcosθ. Substituting for M_U and M_L in eqn 3.5 and integrating, we obtain

$$\delta_v = \frac{1}{EI}\int_0^{\frac{\pi}{2}} (-R\cos\theta))(-PR\cos\theta)Rd\theta$$

$$= \frac{PR^3}{EI}\int_0^{\frac{\pi}{2}} \cos^2\theta d\theta$$

where $\cos 2\theta = 2\cos^2\theta - 1$

$$= \frac{PR^3}{EI} \int_0^{\frac{\pi}{2}} \frac{\cos 2\theta + 1}{2} d\theta$$

$$= \frac{PR^3}{EI} \left[\frac{\theta}{2} \Big|_0^{\frac{\pi}{2}} + \frac{1}{4} \sin 2\theta \Big|_0^{\frac{\pi}{2}} \right]$$

$$= \frac{PR^3 \pi}{4EI} \tag{3.6}$$

For a rectangular cross section, provided that the neutral axis passes through the centroid of the cross-section of the bow spring then $I = \frac{bd^3}{12}$. Where b denotes the width of the rectangle and d denotes the height of the rectangle.

The above derivation involves the introduction of elastic modulus E and the second moment of area I, both of these parameters are needed to calculate vertical displacement δ_v and the required pre-loading force. From the expression of second moment of area, both width and height have dominated the magnitude of the result. The crucial decision to the appropriate size of the spring is the number of features that the spring needs to have. Understanding the requirements of the leaf spring is necessary for determining the size and also second moment of area. The primary function of the leaf spring is to provide both supporting force for the adjacent Steward Platform and tensile force for the nitinol wires. The secondary function is to include a feature for holding and termination of the nitinol wires. Finally both ends of the spring have to be sufficiently thick to allow the insertion of a M2 threaded rod (the threaded rod is for the connection of the ball joint and the bow spring). To meet all these requirements, the final dimension selected was a cross section of 9.5mm x 5mm. The reasons behind these figures will be discussed later.

The final design of the leaf spring not only a function of the above requirements, but also the limitations associated with the ABS properties. A major hurdle consideration is that ABS has a relatively poor heat resistance. This means that the nitinol wires cannot have direct contact with any part of the ABS structure. As the activation temperature of the nitinol wire is 70 - 90°C and the melting temperature of the ABS is 85°C, the only way to avoid this problem was to employ some other material with appropriate good heat resistance to support the wires. Since the bow spring would include a feature for locating this secondary material, the most suitable material for

this job was polytetrafluoroethylene (PTFE). The commonly used name for PTFE is Teflon. This material has many outstanding properties such as – the lowest coefficient of friction (0.05-0.08) of any solid, ability to withstand high temperatures (260°C) and it is almost totally unaffected by commonly used commercial chemicals. These properties were suitable for this application.

Combining all the preceding consideration, the final design requires two additional 9.5 mm x 3 mmϕ Teflon rods. Each of the rods would be located by three pocket features, which were grown as part of the bow spring. To avoid shorting between each loops, each loop of the nitinol wire was separated by two M3 nylon washers from the neighbouring. A schematic view is shown in Fig. 3.7.

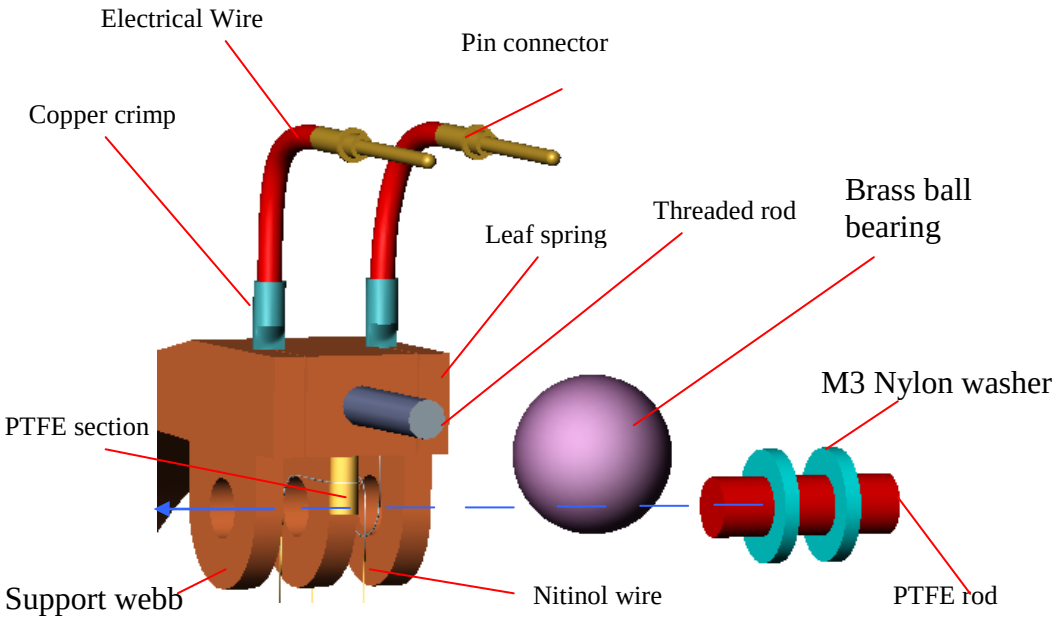


Figure 3-7 A close up view of the end of the spring with the ball joint disassembled.

The nitinol wires were than wound around these two Teflon rods. Each end of the nitinol wires was terminated and connected to electrical wires that were glued (silver loaded epoxy and crimped with copper tube. On the other end of the electrical wires a male pin connector was crimped, to fitt the corresponding female pin connector mounted on the circuit board. These pins connected the nitinol wires to the circuit board. A schematic view is shown in Fig. 3.7. The pin connectors are commercially available from Deutsch Ltd. Refer to Appendix A.

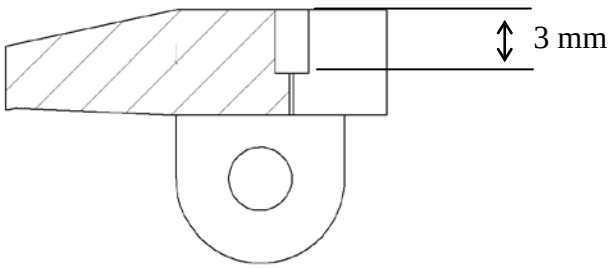


Figure 3-8 Section view of the end of the leaf spring.

With the designed width of 9.5 mm for the leaf spring, it is sufficient to grow two holes of 1.6 mmϕ and with a spacing of 6.1 mm from center to center for locating the copper crimps. At the end of the leaf spring, two blind holes with the initial diameter of 1.6 mm x 3 mm were specially grown for holding the copper tube crimps. A sudden contraction of the diameter was introduced about 3mm from the surface. The diameter is reduced down to 0.2 mm, which is sufficient for the nitinol wire to pass through the remaining thickness of the spring but not the copper crimps. See Fig 3.8. Because of the tensile forces generated by the nitinol wires, the crimps are firmly located in the holes. For assembly/disassembly purposes, the blind holes were cut open along its length until the end of the bow spring, so that the wires can pass through these slots without causing any damages to either the wires or the leaf spring.

Connecting the leaf spring and the spherical joint is a 17.9 mm x 2 mmϕ threaded rod. The design thickness of 5mm is adequate for the spring to hold the threaded rod in place. A simple stress analysis was carried out as follows.

Bearing stresses were induced between the area of contact of the ϕ2 threaded rod and the ϕ2 blind drilled hole of the leaf spring. These stresses were dependent on the area of the contact and the contact force. The bearing stress was obtained by dividing the contact force P by the area of the rectangle representing the projection of the ϕ2 threaded rod on the leaf spring.

$$\sigma_b = \frac{P}{t \cdot d}$$

where t = 10 mm and d = 2 mm.

The bottom deck of the leaf spring would be subject to the greatest stress because of the gravitational force. Therefore the stress at the bottom deck of the leaf spring was required. The maximum contact force occurs when both the tensile force from the nitinol wire and the self weight of the manipulators are combined. The tensile force induced by four 0.003” diameter

nitinol wires would be 3.2 N where by the self weight from the second sections and onward is 0.449 kg. See Appendix B estimation of the self weight. Thus σ_b was calculated as 0.377 MPa. Tensile testing (see next section) gives the yield stress as 15 MPa. The bearing stress is below the limit of the material. So this bearing stress can be ignored.

The spring thickness of 5mm would result in a high stiffness. So over stressing the nitinol could be a problem. To overcome this problem, a loft feature was introduced after the pockets. See Fig 3.9.

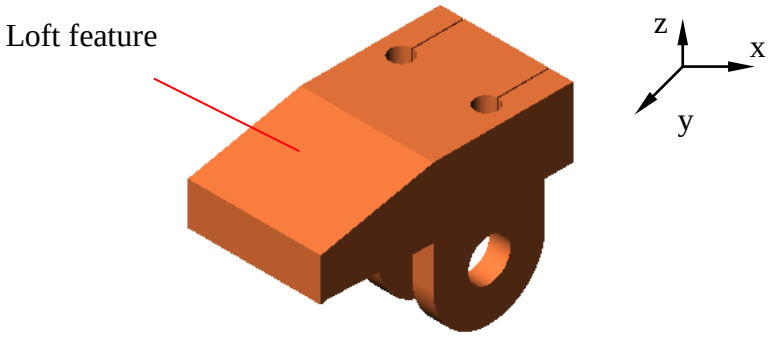


Figure 3-9 A schematic view of the loft feature.

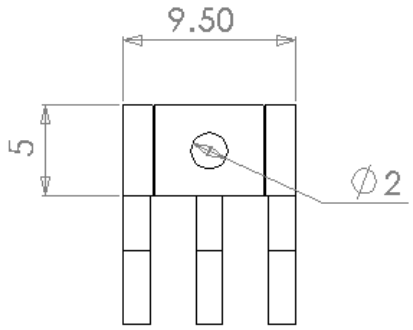


Figure 3-10 Basic dimension of the leaf spring.

In the case of the rapid prototyping process, the loft angle is only introduced along the z-y plane (thickness). See Fig 3.9 for the definition of the axes. Since the rapid prototyping process uses fused deposition modelling, the leaf spring has to be grown with a flat orientation i.e. in the z-y plane. Thus an additional support structure for the support webb is essential. It is difficult to locate any supporting structure for the loft angle if the angle were along the y-x plane. As the leaf spring was lying on the z-y plane and grown along the x-axis, this eases difficulties in the manufacturing process. On top of this benefit, the loft angle also reduces the stress concentration due to the sudden reduction in thickness across the spring. As long as the loft angle reduces the thickness of the leaf spring down to 3 mm (see next section for the working of this thickness), it is not

necessary to have a specific magnitude for the loft angle. Thus the loft angle for the top is 11.9° while the bottom is 2.6° (see Appendix A for more detail).

The material used in the rapid prototyping service is ABS. The prototyping methods are unique in that they add and bond materials in layers to form objects. Through this method, differential air gaps could be introduced within the object. Thus given the same material, the property such as elastic modulus E is no longer the same as models formed by moulding. For this reason tensile tests were carried out as presented in the following section.

3.6.1 Tensile testing

Tensile tests were conducted to determine the mechanical properties of materials. They relate the effect of a uniaxial tensile force to the elongation of a standard specimen. From the specimen geometry both engineering stress and strain can be calculated from the load vs elongation data. Engineering stress (σ) is equal to the force per unit area based upon the original cross-sectional area of the specimen and does not take into account the cross-section decrease as the test progresses. The engineering strain is the elongation divided by the original length of the specimen. Once stress and strain are determined, an engineering stress-strain diagram can be plotted. At stresses below the yield stress, the relationship between stress and strain is linear and follows Hooke's Law. When the stress is removed, the sample returns to its original dimensions. The slope of the linear region of the stress-strain curve is the modulus of elasticity, E, which is usually given in GPa. Unlike other mechanical properties, E is not affected by microstructure but is only affected by the strength of bonds between atoms. Above the yield stress, the elastic limit of the material is exceeded and the material departs from linear elastic behaviour and experiences permanent plastic deformation.

Six tensile specimens of ABS were formed by rapid prototyping. Each of these specimens was about 150mm x 30mm x 1.5mm. Before tensile testing, each specimen was weighed, and from this an average density of 903.703 $\pm$ 50 kg/m^3 was obtained. The specimens were tested with a Hounsfield tensile testing machine. A continuous increasing load was applied to each specimen and the load versus extension recorded with the results being graphed as the test was taking place. The data from the Hounsfield tensile testing machine is summarized in table 3.1 which gives the ultimate tensile stresses and percentage elongations from the six tests done on the ABS specimens. These results are averaged and standard deviation calculated.

Table 3.1 Statistics of Mechanical Properties.

Material	Average σ_y (MPa)	Standard Deviation σ_y (MPa)	Average elongation (mm)	Standard Deviation of elongation
ABS (formed by RP service)	19.32	1.02	2.14	4.35

Figure 3-11 A photograph of the Hounsfield tensile testing machine.

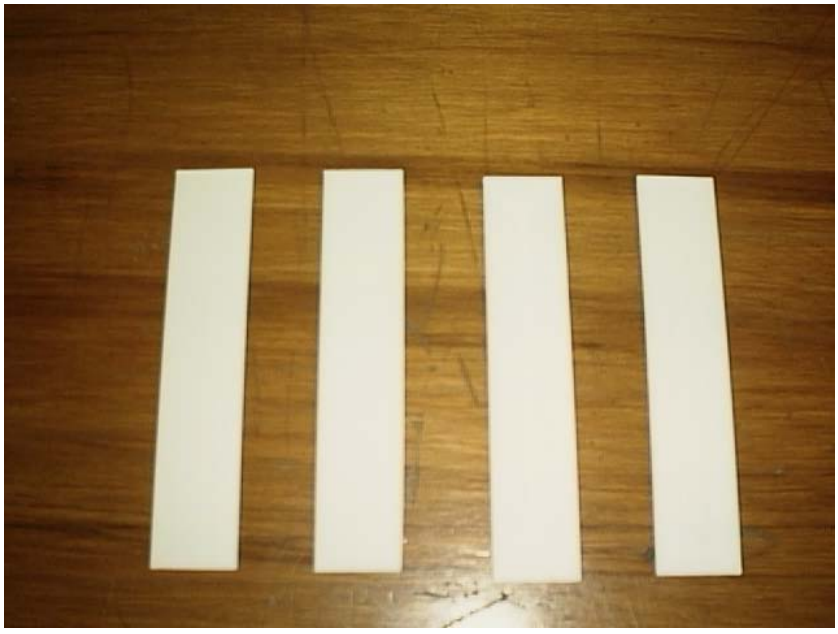


Figure 3-12 The tensile specimens of ABS formed using the rapid prototyping service.

Table 3.2 shows the average values of ultimate tensile stress (σ_{uts}) and percentage elongation of ABS from the tensile tests. The percentage elongation at failure is a measure of the material's ductility. Table 3.2 also shows the standard deviations between these tests.

Table 3.2 Summary of Stress-Strain diagram.

Material	σ_y (MPa)	σ_{uts} (MPa)	E (MPa)	strain (%)
ABS (formed by RP)	19.05	20.1	1395.5	0.0136

All materials have statistical variation in their mechanical properties which must be allowed for when components are designed. The designer must use these variations in such a way that the statistical probability of the component failing in its given application is acceptably low. The material used has a significant impact on the design of components, as different materials show different statistical variations.

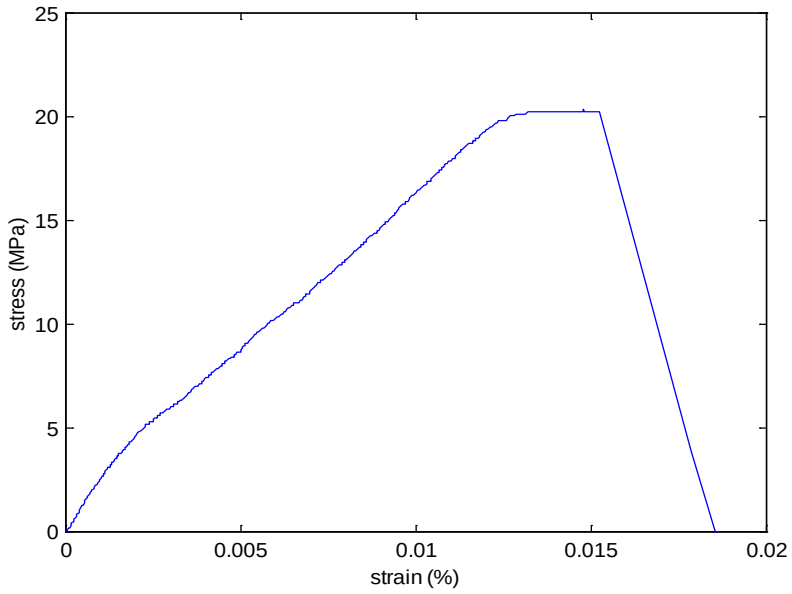


Figure 3-13 A typical graph of stress versus strain for ABS formed by rapid prototype service.

Fig. 3.13 shows a typical stress-strain curve for ABS specimen formed by rapid prototyping. From this curve, the modulus of elasticity of the material can be determined from the linear elastic region which extends up to 19.05 MPa as displayed in Fig. 3.13. The 0.2% offset yield strength, ultimate tensile stress and the percentage elongation at failure can also be found. The average values are summarised in table 3.2.

Using the measured modulus of elasticity in eqn 3.6, the design preloaded force for the nitinol wire and the preload force could be worked out. The technical data from Dynalloy, specifies a maximum tensile force for 0.003" nitinol wire is 0.78 N. Based on this limit, the preload force for

four nitinol wires should not exceed 3.14 N in total. The solution that was obtained from all these limits was that the leaf springs has to have a cross section of 9.5mm x 3mm. Thus the preload force is estimated to be 2N with 5mm displacement. However the design of the leaf spring was not optimised because eqn. 3.7 is not an exact solution for the leaf spring. Furthermore since adding and bonding ABS in layers to form the object, the strength of the product could be closely modelled by treating it as composite. A matlab program has been written for modelling such a composite. Although this program was not utilised, because of fabrication time constraints, the details are given in Appendix C. In order to obtain a better design performance, it is suggested that the leaf spring should be remodelled in this way for any future developments.

3.6.2 Design of the ball joints

Ball joints were used to attach the base platform to the moving platform. They have 2 rapid prototyped ABS hemispherical housing. Each outer hemispherical housing is firmly attached to the platform by two M2 screws. However the problem with rapid prototyping is the surface finish. As previously mentioned the stair case effect from the layering reduces the surface smoothness of an object. This is not acceptable for a ball joint. For this reason each joint was replaced with a 7.9 mmϕ ball made from a brass ball bearing, which has a better surface finish than a rapid prototype ball joint. Since the ball bearing and the leaf spring were no longer manufactured as a single piece, a 3.95 mm deep blind hole was drilled and tapped to accept a M2 x 17.9 mm threaded rod. The threaded rod connects the ball bearing to the leaf spring. Excluding the features of the stem and the ball joints, the leaf spring becomes a much simpler object to rapid prototype. To ensure the threaded rod can withstand the stress of the platform weight, a simple stress analysis was carried out.

Excluding the immobile base Platform and the first section of the springs, the weight of the other 3 manipulators including 18 leaf springs, 48 brass ball bearings, 48 of 2 mmϕ x 17.9 mm threaded rods, 36 of ϕ3 x 9.5 mm Teflon rods, a base cover and 7 prismatic triangular boxes was estimated to be 0.58 kg. This was using the assumptions that the entire weight of the manipulators was acting as a point load at the tip of the 2 mm diameter threaded rod with a maximum moment arm of 17 mm.

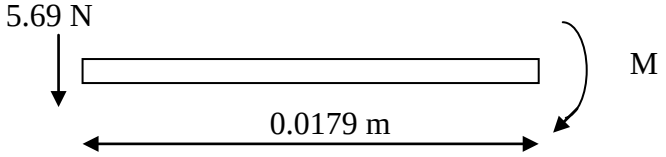


Figure 3-14 A free body diagram for the 2mm diameter threaded rod.

$$M = Px = (0.58 \text{ x } 9.81) \text{ x } 0.0179 \quad (3.8)$$
$$= 0.102 \text{ Nm}$$

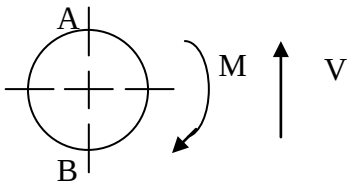


Figure 3-15 A free body diagram on the cross section of the 2 mm diameter rod.

The second moment of area I for a circular rod is calculated as:

$$I = \frac{\pi d^4}{64} = \frac{\pi(0.002)^4}{64} = 7.85 \text{ x } 10^{-13}\,\text{m}^4 \quad (3.9)$$

The bending stress is worked out as:

$$\sigma_b = \frac{My}{I} = \frac{(0.102)\cdot(0.001)}{7.85x10^{-13}} = 0.13 \text{ GPa} \quad (3.10)$$

Shear stress in bending:

$$\tau_{average} = \frac{V}{A} = \frac{5.69}{\pi(0.001)^2} = 1.81 \text{ MPa} \quad (3.11)$$

and the theoretical yield stress for ASIS 1040 steel is 415 MPa. Thus the 2 mm diameter threaded rod is capable of withstanding both shear and bending stress. Moreover the bending stress would be distributed equally by six of these rods. Therefore there should not be any problem on the threaded steel rods.

The elevation angle is given as below:

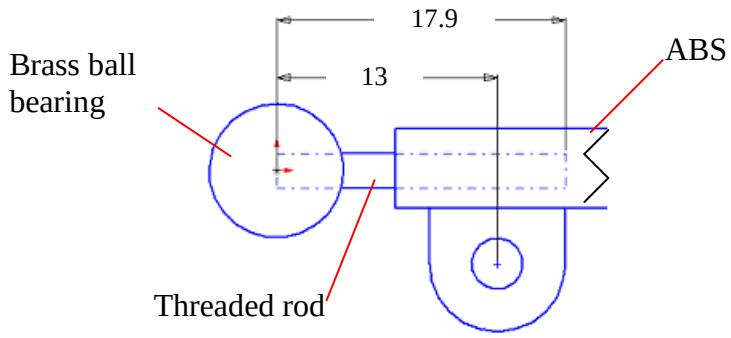


Figure 3-16 A simplified view of the ball joint.

From Fig. 3.16, the distance between the center of the ball bearing to the center of the Teflon rod is 13 mm. The length for nitinol wire to complete one loop of winding between both Teflon rods is 90 mm. With 5% contraction of the nitinol wire, the stem would roughly rotate by ± 19.1° about the ball bearing. Although the ball joint used in the Stewart Platform needs to achieve only a small elevation angle, the elevation angle was designed to be 30° (See Fig. 3.17 below for the definition of the elevation angle). The extra 10° is to compensate for the buckling effect caused by self-weight. As it is expected the bottom module of the leaf spring would suffer highest buckling force compared to the top module. Thus the ball joint could potentially achieve an extreme elevation angle of about 30°.

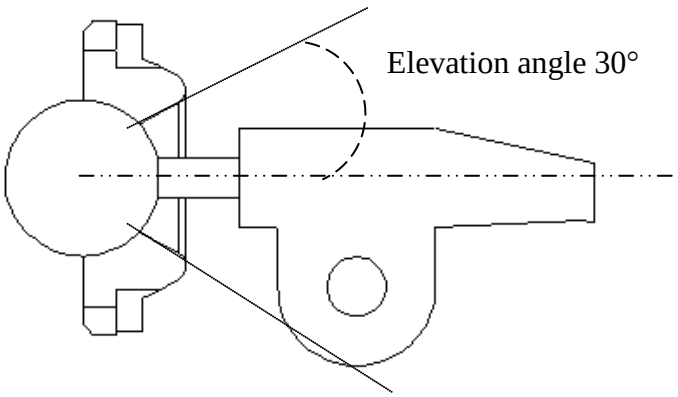


Figure 3-17 Part cut away of a ball joint cone.

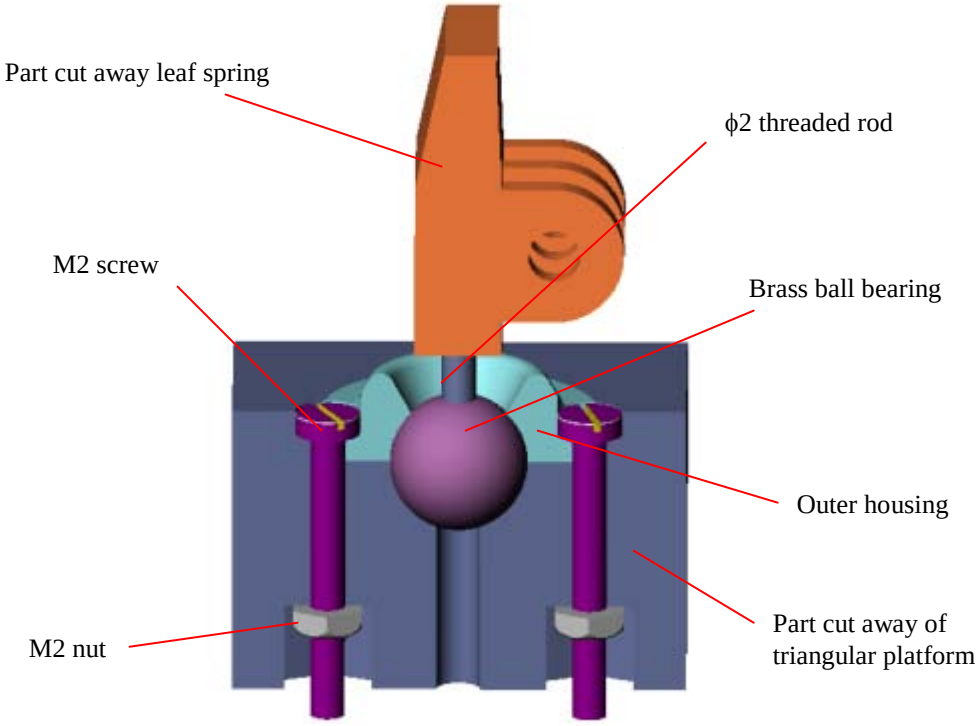


Figure 3-18 Part cut away of a ball joint.

3.7 Static analysis

To ensure that the structure was statically stable, a static analysis was carried out. This analysis only focus as on the bottom stage of leaf spring because it experiences the greatest stress induced by the weight of the mechanism. This means that this area is the critical part of the mechanism. In this case, the weight of the lower prismatic equilateral triangular box and the base cover are not taken into account because they are stationary and sitting on the floor. Fortunately, weight is equally supported by six leaf springs so the structure should not be too bulky even with the extra thickness for rigidity.

The leaf spring is lying in z-x plane, defined as in Fig. 3.19. The compressive force caused by the mechanism's weight acts in the z-x plane. Thus in order to resolve the compressive force P into two components, some angles need to be determined (see Fig 3.20).

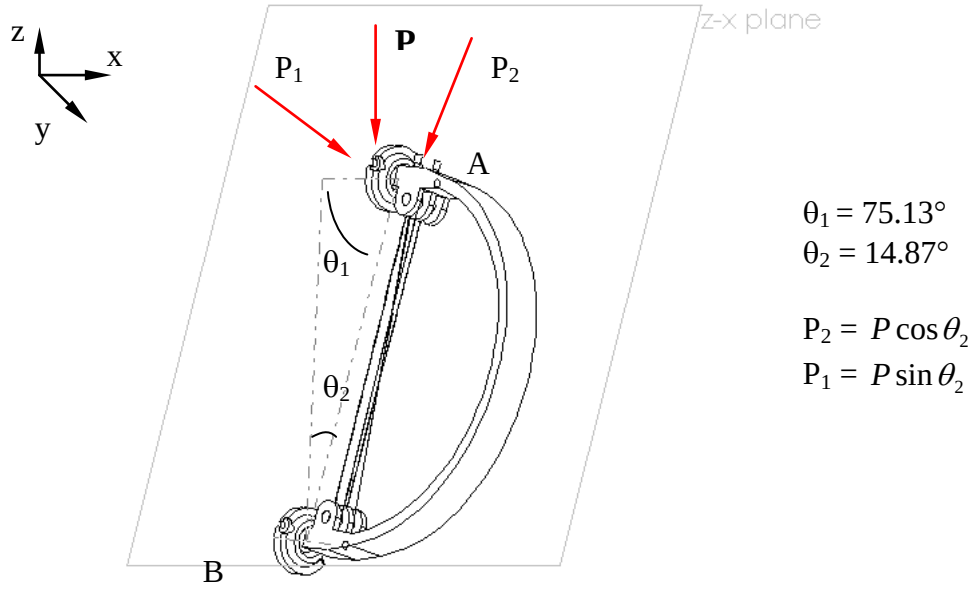


Figure 3-19 Plane justification for a leaf spring.

First consider the internal actions subjected to the z-x plane loading (See Figs. 3.19 and 3.20).

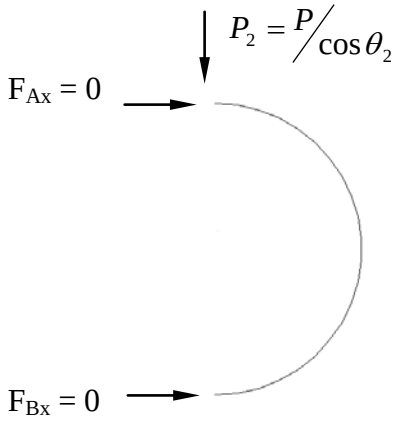


Figure 3-20 Free body diagram of a simplified leaf spring.

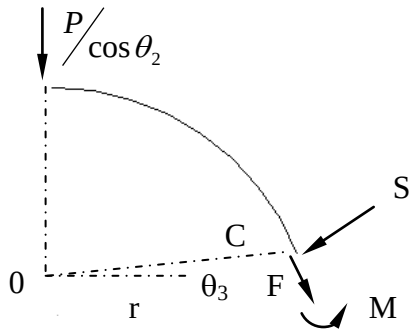


Figure 3-21 A sectioned free body diagram.

The compressive force P_2 can be calculated by excluding the lower part of the prismatic equilateral triangular box and the base cover, which leaves the rest of the structure's weight (0.562 kg). Each leaf spring would need to support ≈ 0.094 kg (See Appendix B for the details of the weight estimation).

The internal forces were calculated by applying the methods of sections to obtain the equations of equilibrium as follows:

Resolving radially: $$S + P\cos\theta_2 \cos\theta_3 = 0 \qquad (3.12)$$

$$S = -P\cos\theta_2 \cos\theta_3 \qquad (3.13)$$

Resolving tangentially: $F + P\cos\theta_2 \sin\theta_3 = 0$

$$F = -P\cos\theta_2 \sin\theta_3 \qquad (3.14)$$

Taking moments about O gives:

$$M - Fr = 0 \qquad M = Fr = -P\cos\theta_2 r\sin\theta_3 \qquad (3.15)$$

As expected the most critical point of loading is when $\theta_3 = 0°$ and $\theta_3 = 90°$. When the forces and moment are:

S = 0.0906 N F = 0 N M = 0 Nm

Thus shear stress is:

$$\sigma_s = \frac{S}{A} = \frac{0.0906}{2.85 \times 10^{-5}} = 3.179 KPa \quad (3.16)$$

When $\theta_3 = 90°$

S = 0 N F = 0.0906 N M = 0.00408 Nm

Thus the compressive stress and bending stress are:

$$\sigma_s = -\frac{F}{A} = -\frac{0.0906}{2.85 \times 10^{-5}} = 3.179 KPa \quad (3.17)$$

$$I = \frac{bd^3}{12} = \frac{(0.0095) \cdot (0.003)^3}{12} = 2.1375 \times 10^{-11} m^4 \quad (3.18)$$

$$\sigma_b = \frac{My}{I} = \frac{0.00408 \cdot 0.0015}{2.1375 \times 10^{-11}} = 0.286 MPa \quad (3.19)$$

The torque induced by force P_1 is calculated as below:

$$T = P_1 r = P \sin\theta_2 \cdot r = 0.00108 Nm \quad (3.20)$$

The torsional shear stress for a rectangular cross section is:

$$\tau = \frac{T}{c_1 bd^2} \quad (3.21)$$

where c_1 is a constant equal to 0.267[13], Thus:

$$\tau = 47.3 kPa$$

Since several different stresses occur simultaneously, it is necessary to study the resultant stresses acting at certain critical points. In particulary points A and B were chosen as it is expected that these are the critical points,when $\theta_3 = 0°$.

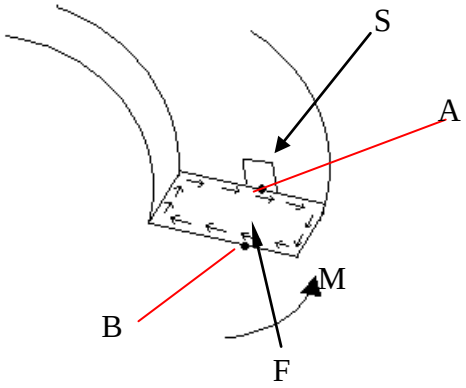


Figure 3-22 Combined stresses at point A and B.

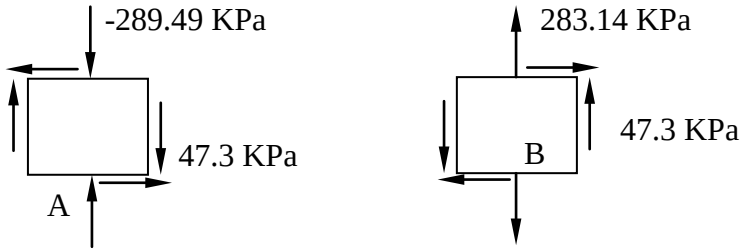


Figure 3-23 Resultant stresses at point A and B.

Square elements located at A and B on the surface of the leaf spring were used as shown in Fig. 3.23 to summarise the results obtained. Compared with the yield stress, the self-weight would not cause any problem with the lower part of the leaf springs.

3.8 Conclusion

Fused Deposition Model fabricates objects by forming ABS in layers. Through this method, differential air gaps would introduce within the object. Thus the object's properties would be different when formed with injected moulding. To understand the new properties, six tensile testing samples were fabricated by Rapid Prototype and were tested with the Hounsfield tensile testing machine. The yield strength, ultimate tensile strength and elastic modulus were determined.

Stress analysis was carried out for the leaf spring and M2 thread rod. This was to ensure the Steward Platform actuators were robust enough to support the weight of the structure, as well as to ensure the ABS used in Rapid Prototyping was a suitable material for the entire structure. The result was that the Rapid Prototype parts are light enough for the

SMA wires to actuate the Stewart Platform, but was also strong enough to produce rigid structure.

Chapter 4 Testing Methodology and Results

4.1 In search of testing method

The cyclic behaviour of SMA is important in practice. Various methods were used to investigate these properties and they are described in this chapter.

Since the resistance of the nitinol wire is dependent on both the temperature and the strain, the experimental set-up must be able to vary these two variables.

$$R=R(\varepsilon, T)$$

Initially, the nitinol wire had been tested in both an oven and a water bath. The specimen was a TI-Ni SMA wire, 0.762 mm in diameter and 113 mm in length produced by Dynalloy, Inc (see Appendix E for the properties of the nitinol wire). One end of the nitinol wire was tensioned with a small metal spring with a stiffness of 10 N/mm and the other end of the wire was constrained with a brass clip. The brass clip was hooked to a metal pin, mounted on a Perspex board. The whole Perspex board was placed in the water bath. During testing the change in length of the wire was measured manually with a digital vernier, and the resistance of the wire was measured using an "Automatic LCR meter" type 4225. Each length reading was repeated several times, in order to minimise parallax errors. However some difficulties become evident. First, the water bath tank could not reach a temperature of 80° C. This problem wassimply overcome by placing an additional heating element inside the water bath tank but this means that the temperature inside the water tank can only be controlled manually.

Although this is tedious work, the main problem is that this action also generates electrical noise that interferes with the measurement of resistance. Furthermore it was difficult to adjust the bath to the desired temperature. If the bath was too hot it was difficult to lower the temperature to the desired level. Then if temperature of the bath dropped too low it then has to be adjusted upwards. It then would rise above the desired temperature and have to be adjusted down again. Eventually this forms a minor hysteresis temperature loop inside the nitinol wire while determining the hysteretic behaviour of the nitinol. Consequently the data collected does not represent the true behaviour of the test sample and also affects the lifetime of the nitinol wire. Furthermore, the water tank was not one complete piece. A missing part is believed to provide coolant to the water bath. Thus, there is no coolant running in the water tank, so the time for the water bath to cool down is rather long. Eventually this test method was abandoned.

The use of an oven was then investigated. The advantage of using an oven is that the temperature can be automatically adjusted to a manually preset temperature. Thus, the problem of forming a temperature loop can be avoided. Nevertheless, the data that was collected through the use of the oven does not correspond well to the relevant textbook data. It shows that there is a lot of noise generated by the oven. This is mainly due to the way the oven controls the temperature. The oven switches on and off automatically whenever it detects that the temperature is not at the desired temperature. This switching generates electrical noises. It is still time consuming to set the desired temperatures manually, and the cooling period was quite lengthy. One way to get around this problem was to introduce another variable - time. The oven was turned full on and the oven and the nitinol wire were heated from room temperature for 2 hours. Once the oven had heated up for a period of time, the oven was then turned off completely to cool down the wire. Readings for this whole hysteresis process were taken. Eventually the collected data would be in terms of time. Even though the switching action had been minimised, electrical noises still exist. See Fig. 4.1.

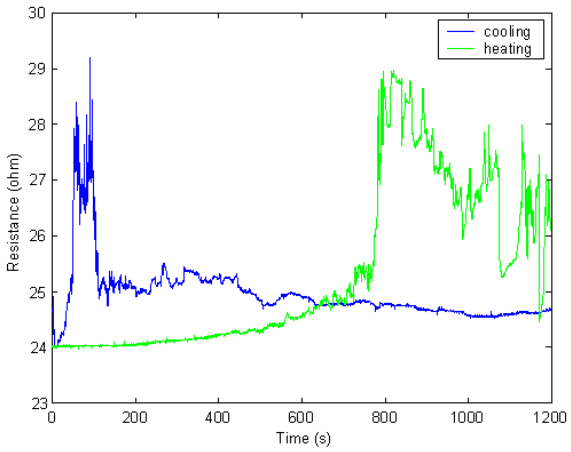
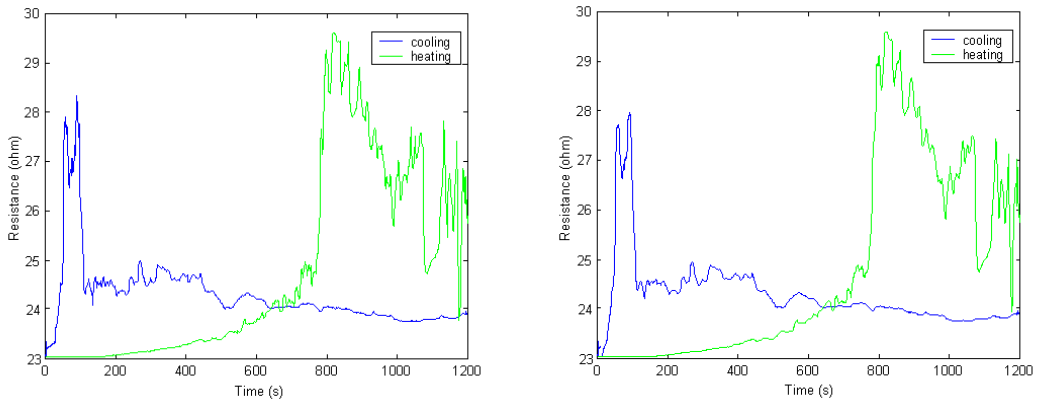


Figure 4-1 Resistance of nitinol wire over time tested in an oven at constant rate of heating.

A clearer relation of resistance-time characteristic was needed from this set of raw data. Thus boxcar filtering was introduced. See Figs. 4.2 and 4.3. An averaging of every 20 points and 50 points were plotted. The influence of electrical noise was still present so this data set was discarded.



(a) (b)

Figure 4-2 Boxcar filtering for the resistance-time characteristic of nitinol actuator. (a) Boxcar filtering of 20 points; (b) Boxcar filtering of 50 points.

4.2 Testing with resistance monitoring circuit

Both methods previously mentioned were considered to be unsuccessful. However the use of a modified Wheatstone bridge to analyse the resistance of the wire was suggested [14]. A basic circuit layout is shown in Fig 4.3.

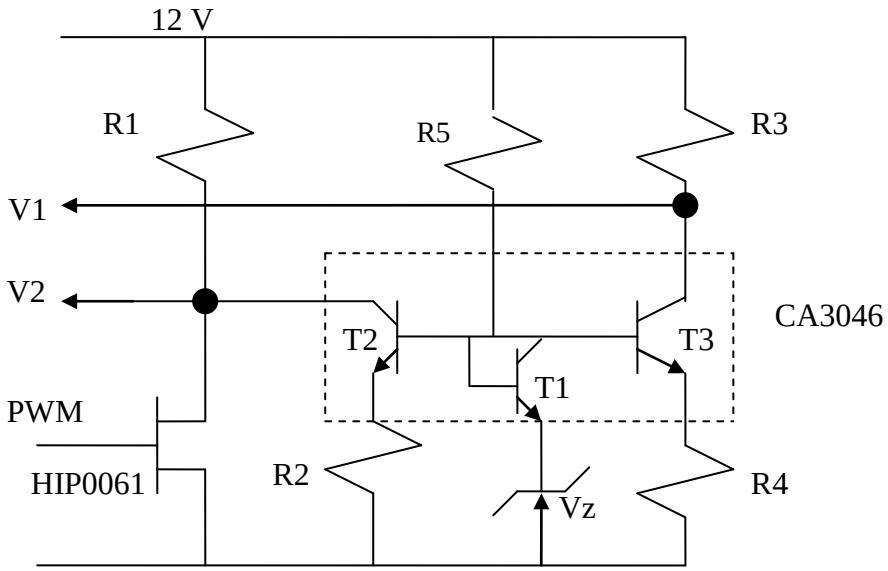


Figure 4-3 The PWM driver and resistance monitoring circuit.

T1 is wired as a diode in order to compensate for the voltage – temperature variations in base – emitter junctions of the other transistors. They were on the same substrate thus providing a good temperature match, and the circuit operates as a current mirror. Details of the electronic components are shown in Table 4.1.

Table 4.1 List of electronic components.

Electronic Component	Code number
Quad Transistor Pack	CA3046
FET	BUK 102-SOGS
Resistor R1	Nitinol Wire
Resistor R2	570 Ω
Resistor R3	70 Ω
Resistor R4	1.017 k Ω
Resistor R5	3.3 k Ω

This differs from direct measurements in that the quantity being measured is compared with a known reference quantity. This strategy avoids unwanted interaction effects and usually results in greater precision than the direct measurement. By driving the wire with different PWM values, the temperature is varied. The voltage difference $V_1 - V_2$ is measured during the off time of the PWM. The nitinol wire was mechanically attached between a peg and a tensioning wire, and electrically attachments were connected across the nitinol wire only. So photographs of the apparatus are shown in Figs 4.17 and 4.18. In addition the tensioning wire was also curled around an encoder, so that From this the strain of the wire can be detected. Another outstanding feature was that the Wheatstone bridge was linked up to a computer, which has RTX installed for real-time data collection. Therefore both resistance and strain can be carefully recorded. The Wheatstone bridge was used for the testing and the results were found to closely match the results reported in the literature [3]. See Figs 4.5 and 4.6 for typical result.

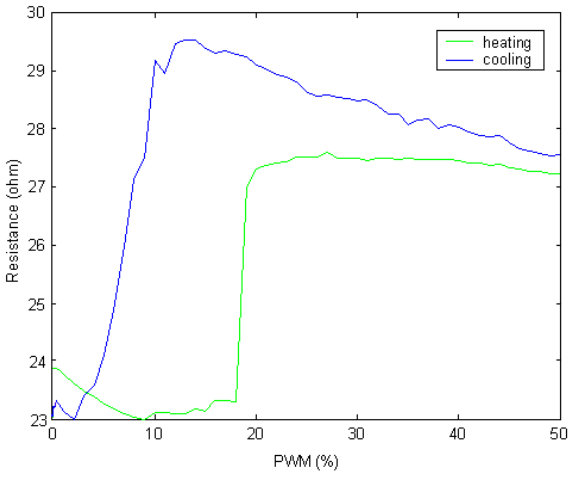


Figure 4-4The resistance- PWM characteristic of nitinol wire at constant tension at 1N.

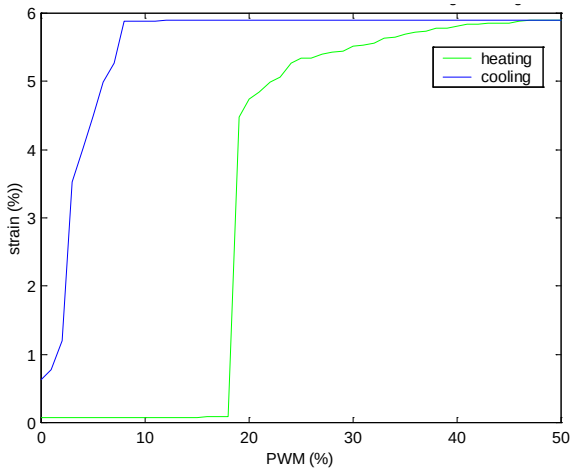
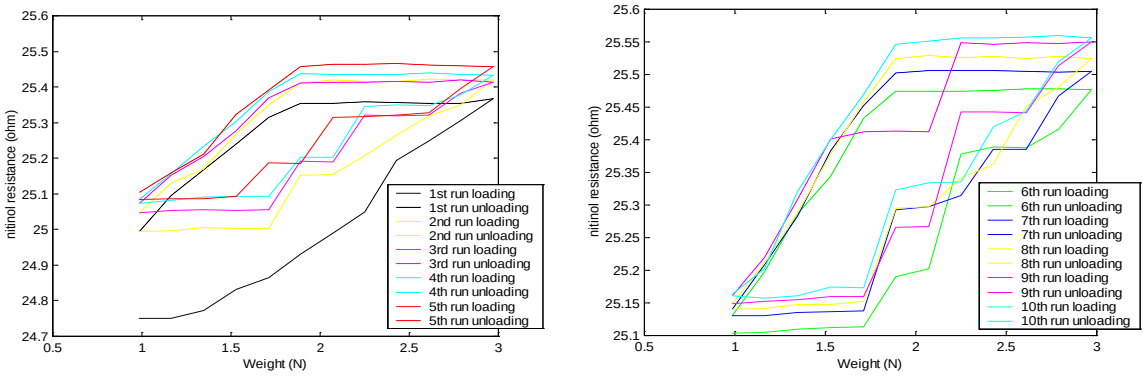


Figure 4-5 The strain – PWM characteristic of nitinol wire at constant tension of 1N.

With this arrangement, testing of the wire strain effect was performed for various loads at a room temperature of 18.6°C. The load was increased incrementally, and then both resistance and strain were recorded. This procedure was repeated until just before the wire exceeded its yield point, and then the wire was unloaded incrementally. The material specification provided by the supplier of nitinol stated that it has maximum pulling force of 0.8 N for wire size of 0.03" in diameter. Thus testing was carried out so as not to exceed the critical load. The material specifications for the nitinol wires are listed in Appendix E.

Eventually this whole process was repeated a number of times to establish the repeatability of the loading cycling. The results were repeatable, so the wire showed potential as a useful actuator, one of the aims of this research. Care was taken not to introduce shock loading while changing the load incrementally (adding or removing weights).

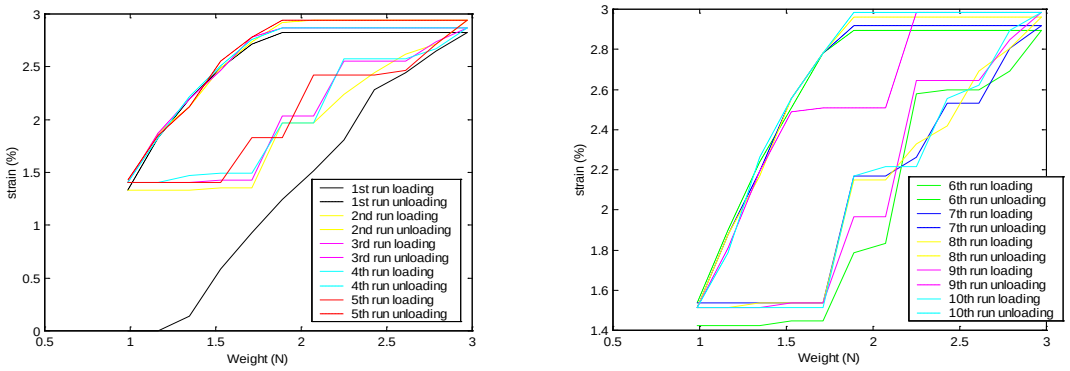


(a) (b)

Figure 4-6 The resistance - load characteristic of nitinol wire subjected to cyclic loading. (a) runs 1-5; (b) runs 6-10.

Fig 4.6 shown that as each run progress along, the hysteresis loop shifted upwards slightly. This effect is due to the specimen has yet to be "trained" (see section 4.5 for more explanation about the

phenomenon of training). The starting of the first run, which is coloured in black, was happened to be out of the main hysteresis loop. This is not surprised as the specimen is starting to behave like hysteresis loop behaviour. Once it has established the hysteresis loop the starting line was disappeared in the later runs. All of the runs have shown that the trend of the hysteresis loop was about the same.



(a) (b)

Figure 4-7 The strain - weight characteristic of nitinol actuator subjected to cyclic loading. (a) runs 1-5; (b) runs 6-10.

Figs 4.7 (a) and (b) shown the output strain of the nitinol wire was also undergo hysteresis loop behaviour. The range of the strain lies between 1.5-3%.

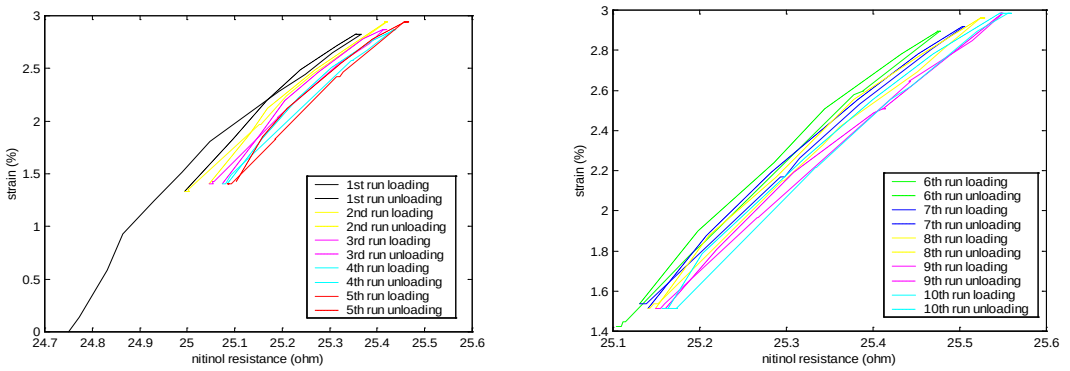


Figure 4-8 The strain - resistance characteristic of nitinol actuator subjected to cyclic loading. (a) runs 1-5; (b) runs 6-10.

Results from Figs 4.6 and 4.7 are combined to yield Fig 4.8. Again a repeatable hysteresis loop was obtained. The shifting phenomenon is expected to be settle in later runs.

4.3 Alternative testing method with heating element

The next dependent variable is temperature. A slight modification to the thermal isolation of the wire from the ambient temperature was carried out in comparison with the previous set up. A polystyrene box was built with an additional layer of thermo blanket attached to the inner side of the box. The

operation temperature inside the box was set with a heating element and a small fan placed near to the wire. In the ease of drawing out heat, a small hole is drilled at the back of the polystyrene box for mounting a secondary fan. This arrangement allowed raising and dropping the operating temperature of the wire easily. The whole experiment ran continuously for 18 hours (up to 10 cycles.) A schematic of the experimental layout is shown in Figs 4.9 and 4.10.

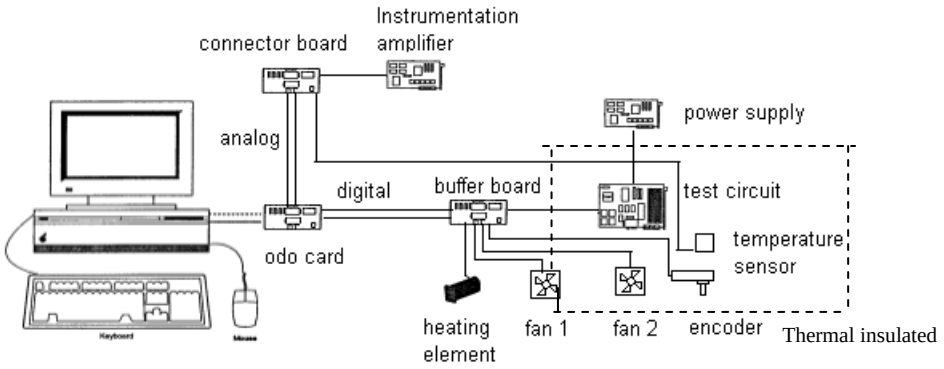


Figure 4-9 An experimental set up of heat cycle test.

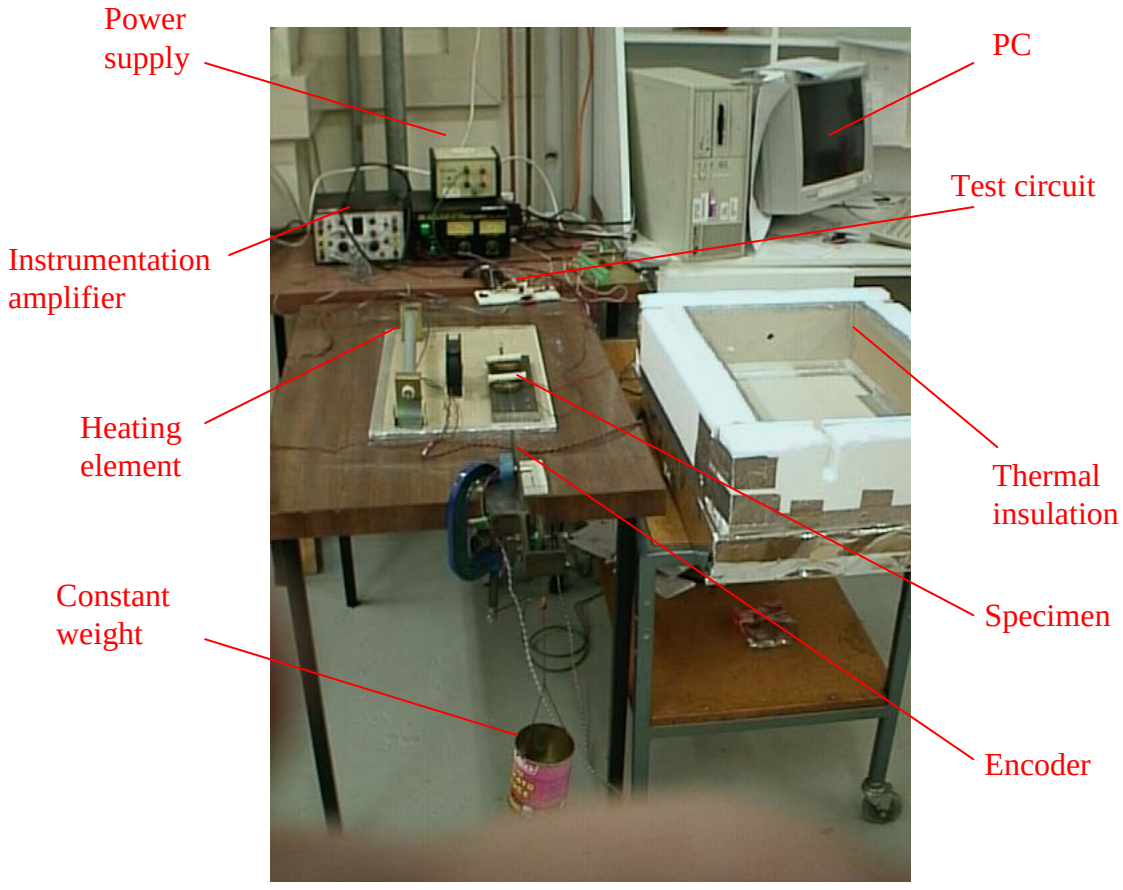


Figure 4-10 An actual experimental set up of heat cycle test.

This whole process was repeated a number of times to establish the repeatability of the thermal cycling. The testing process was monitored and recorded using a computer. The results were repeatable so the wire again showed potential as a useful actuator, one of the aims of this research. See Figs 4.11, 4.12 and 4.13 for typical results.

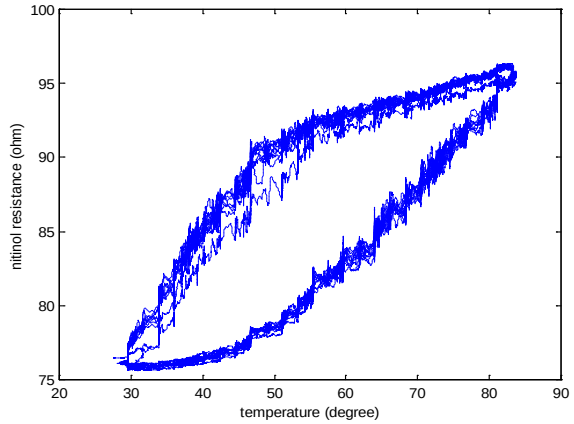


Figure 4-11 The graph of resistance-temperature characteristics of nitinol wire externally heated and cooled in many cycles.

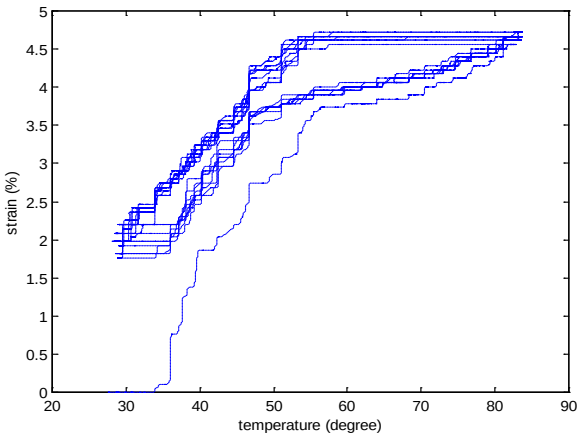


Figure 4-12 The graph of strain-temperature of nitinol wire externally heated and cooled in many cycles.

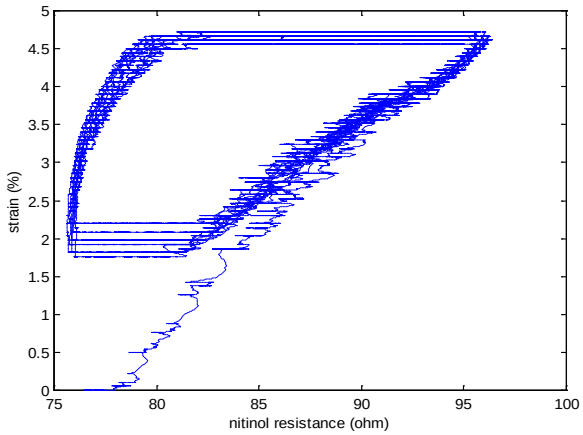


Figure 4-13 The resistance-strain of nitinol wire externally heated and cooled in many cycles.

Although the results are noisy, the overall trend can still be visualised in the three graphs. From Fig 4.11, the long-term reliability electric resistance and force of the SMA wire has been plotted after first undergoing many cycles of "training". The other graphs show the initial characteristics at the start of the cycling. SMA wire showed excellent characteristics both in terms of generated force and electric resistance. Fig 4.13 shows the stable relationship between resistance and strain which means that the wire could be used in a wide range of applications as a highly reliable actuator that can incorporate feedback control.

4.4 SMA Actuator Performance Testing

All of the previously mentioned work involved testing the characteristics of the nitinol wire. One issue of the primary concern to a designer was whether the final design of the SMA actuator would alter the characteristics of the nitinol wire. More importantly, tests should be made to ensure that the spring stiffness does not degrade noticeably after prolonged use; in other words, the reliability and durability of the SMA wires were a major concern. For these reasons a leaf spring was tested with the use of the same apparatus previously mentioned in section 4.3. photographs of the apparatus are shown next.

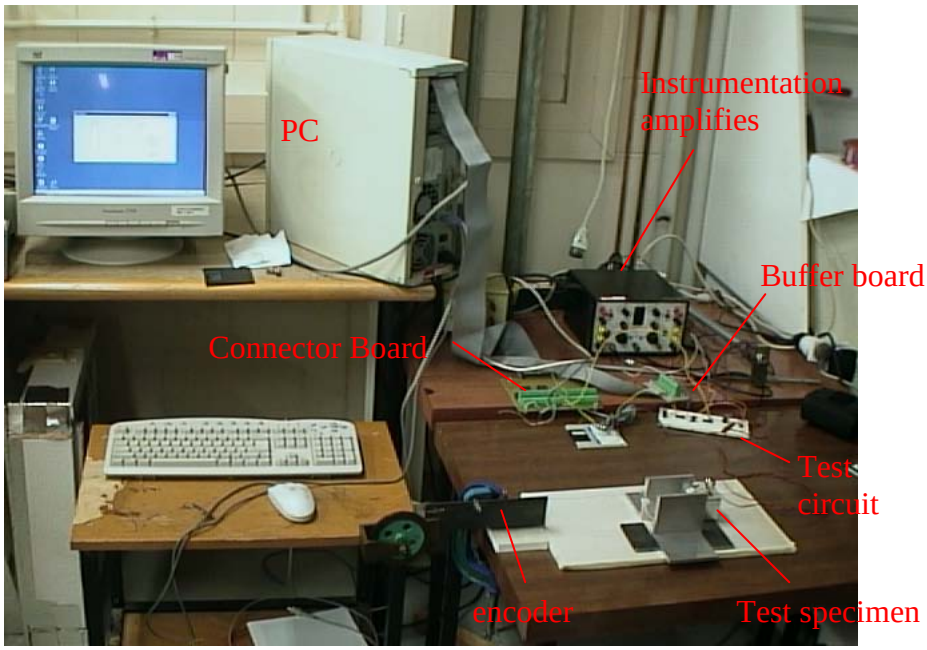


Figure 4-14 An experimental set up for the actuator.

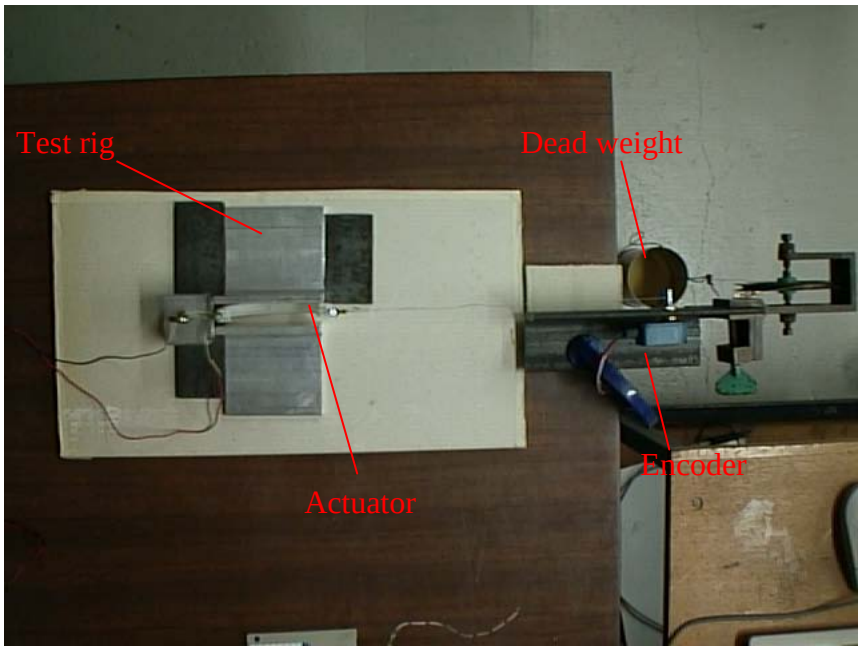


Figure 4-15 Top view of the test rig.

In order to have a better understanding of the characteristics of the SMA actuator, several experiments were conducted using various approaches. The first method was to heat the nitinol wire (which is threaded and tensioned in the leaf spring) by electric current in PWM signal form, and cooled by natural convection. Using the modified Whetstone bridge and encoder, the wire resistance and the corresponding strain were recorded. This first experiment provides a simple means of evaluating the range of the wire resistance and the percentage strain. To ensure that the data collected through this method was repeatable, a second method was introduced. The second method involves heating and

cooling the nitinol wire repeatedly for ten times. If the characteristics remains the same, the results would hence be deemed repeatable. A final method was employed to check the effects of the minor hysteresis loop. The method involves evaluating the existence of the minor hysteresis loop that prevents the nitinol wire from transforming either to fully austenitic phase or fully martensitic phase during the experiment. This was achieved by initially adjusting the PWM signal to a maximum percentage then lowering the PWM signal to a certain percentage but not totally turned off and consequently reheated again. Alternatively, adjusting the PWM signal from zero to a certain percentage that is insufficient for the temperature in the wire to fully cause the austenite phase transformation. The PWM signal was then lowered zero percentage again. Both ways would generate reduced hysteresis loops within the nitinol wire.

All three experimental methods were applied during testing and were under the same experimental conditions; i.e. the leaf spring was attached to the load of 1N. This constant load was chosen such that the nitinol wire is stretched when it cool but lifts the load when heated. The test specimen was a leaf spring threaded with 0.762-mm in diameter a nitinol wire and 345 mm in length. This particular wire length was chosen as the final desired length for every leaf spring and give 4 strands acting in parallel. From these conditions, the data collected should characterise the assembled leaf springs.

Figs 4.16, 4.17 and 4.18 illustrate a typical characteristic of the nitinol wire using the first approach. The results showed that the transformation temperatures are reached between 15% to 40% PWM, with a maximum strain of 4.5%. More importantly was that the shape of the hysteresis loop remains more or less the same in comparison with the result shown in Fig 4.11. Although the shape of the hysteresis loop remained the same, Fig 4.16 showed clearly that the hysteresis loop has shifted to the right. This is due to the pretension force caused by the leaf spring, which causes stress-induced martensite as was discussed in Read chapter 2.

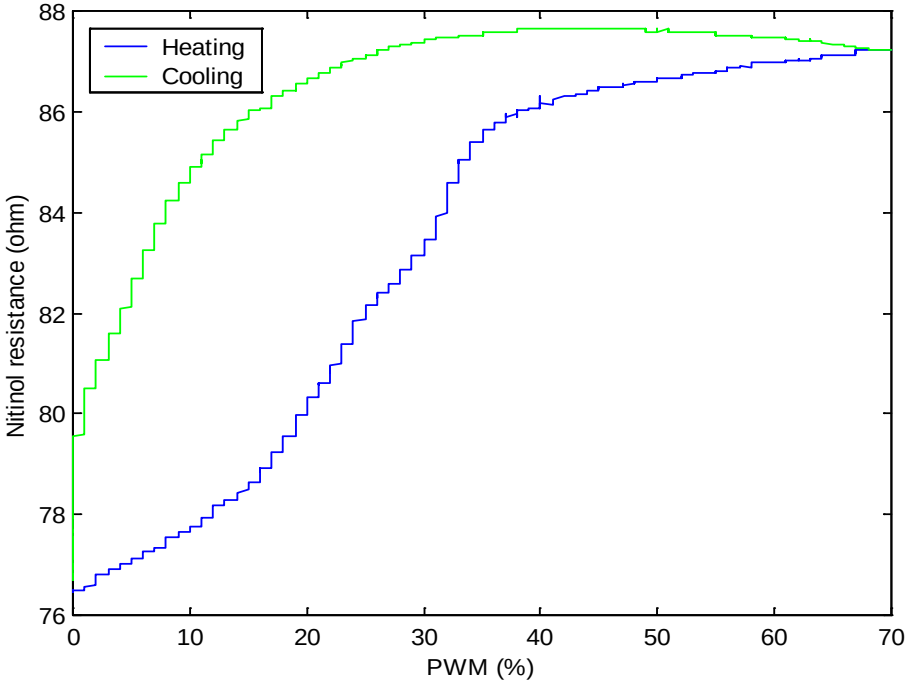


Figure 4-16 The graph of wire resistance-PWM characteristics of the SMA actuator.

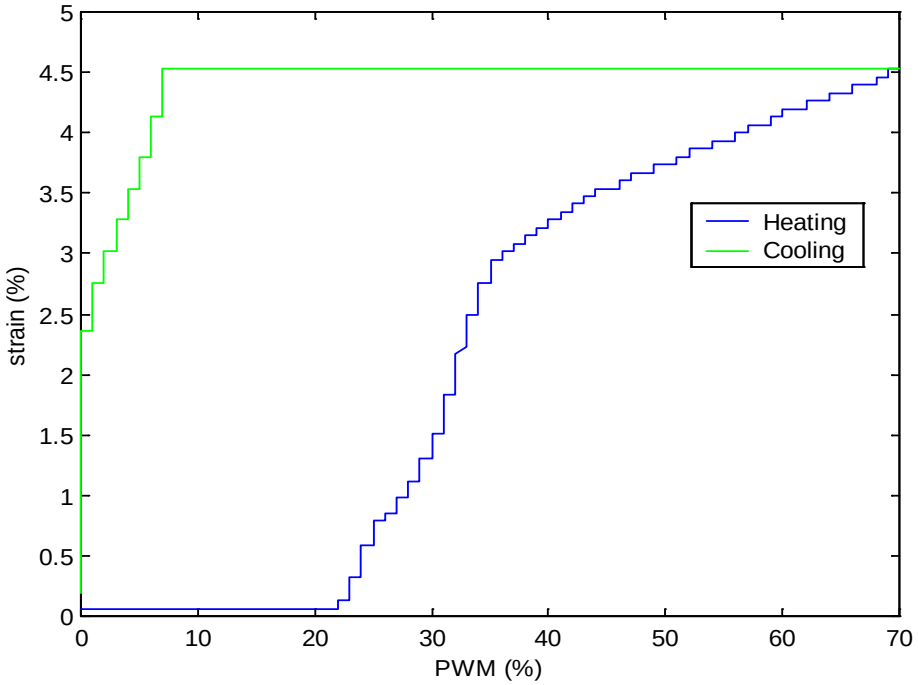


Figure 4-17 The graph of strain -PWM characteristics of the SMA actuator.

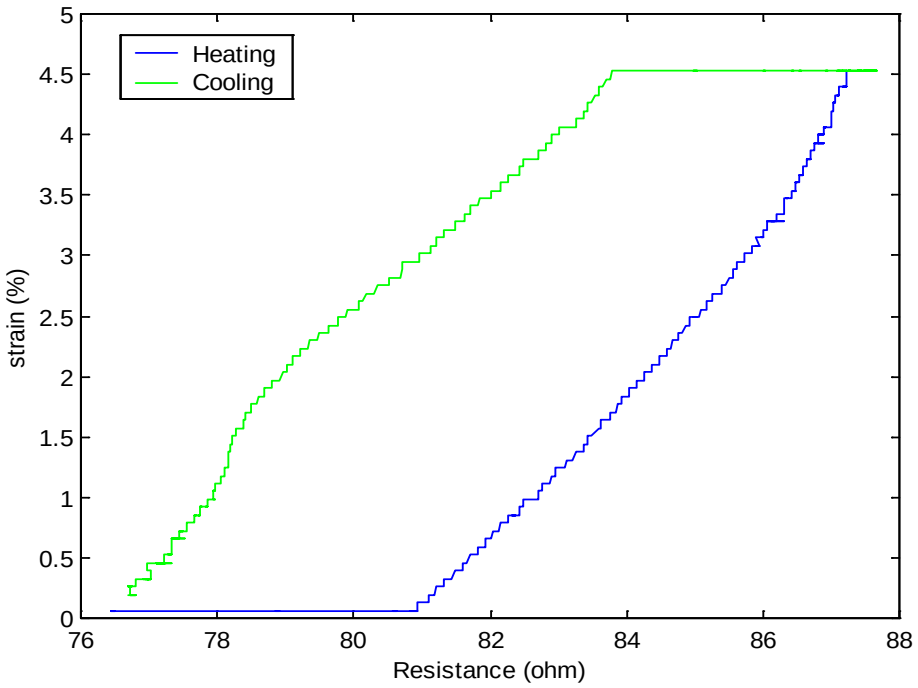


Figure 4-18 The graph of resistance –strain characteristics of the SMA actuator is obtained by combining the measured results from the previous two graphs.

To gain more precision, each assembled leaf spring was tested as well. The results were summarised as follow. The plots of each individual leaf spring are presented in Appendix D.

Table 4.2 Summarised results for all the 24 actuators.

Leaf spring No.	Resistance (ohm) at 0% PWM	Resistance at 70% PWM
10	76.51	87.8082
11	76.71	88.0567
12	76.81	87.2259
13	76.873	87.1027
14	76.65	87.8704
15	76.47	87.2568
20	76.53	87.2461
21	76.12	87.9543
22	76.25	87.8082
23	76.55	87.1969
24	76.25	86.3458
25	76.84	87.779
30	76.83	88.4528
31	76.7	87.1099
32	76.3	86.3037
33	76.05	86.0741
34	76.23	86.6185
35	76.55	87.8392
40	76.57	88.4215
41	76.25	87.4875
42	76.65	87.7224
43	76.53	87.4566
44	76.54	87.0447
45	76.57	87.6914

Figures 4.18, to 4.23 illustrated the typical initial cyclic characteristics of the nitinol wire using the second approach. They are presented in blocks of 5 cycles for clarity. The results showed that, for up to ten runs, the characteristics of the assembled leaf spring remain unchanged. Nevertheless more test are needed in order to confirm that the leaf springs are durable and the characteristics are repeatable.

In the ideal case, all the assembled leaf springs would have the same performance. i.e. each nitinol wire would pull with equal force. In practice this does not occur for two reasons. The major reason because of the actuator construction. There are slight variations in the length of each piece of nitinol wire. The tolerance is expected to be ±1mm. The cause of this unequal length is due to the difficulty in tying the second termination knot in the precise location to join the nitinol wire to the upper electrical supply wire. The second reason is that, as the nitinol wire is being threaded into the leaf spring, it is likely that the tension in the nitinol wire is not uniform. Since these errors are quite small they are acceptable. Unfortunately these errors can only be minimised, but never eliminated.

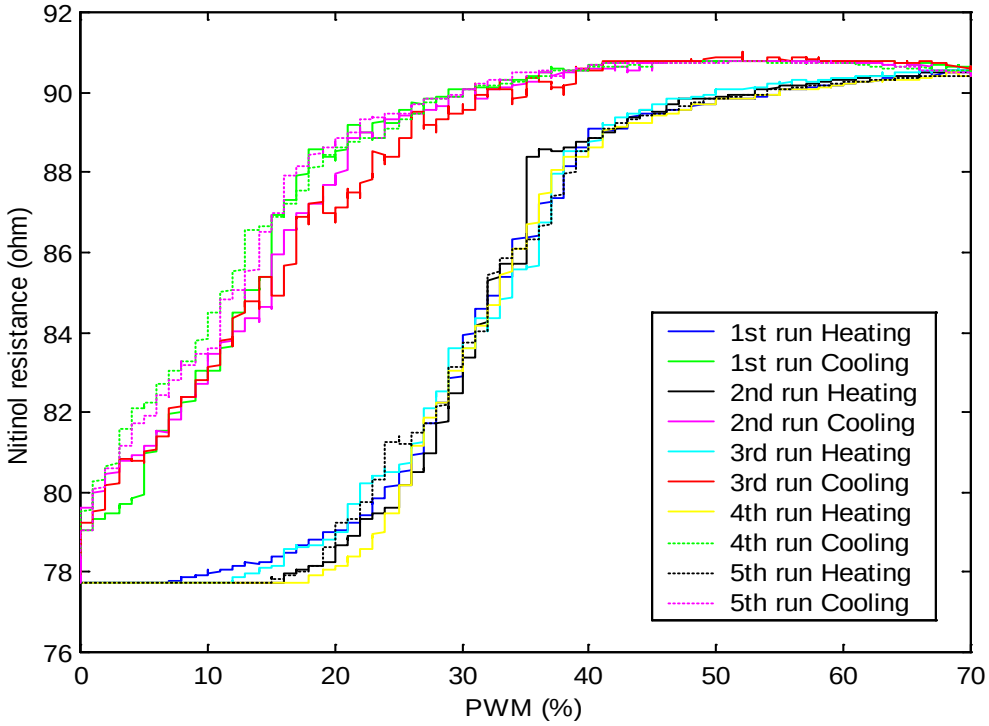


Figure 4-19 The graph of wire resistance –PWM characteristics of the SMA actuator (runs 1 – 5).

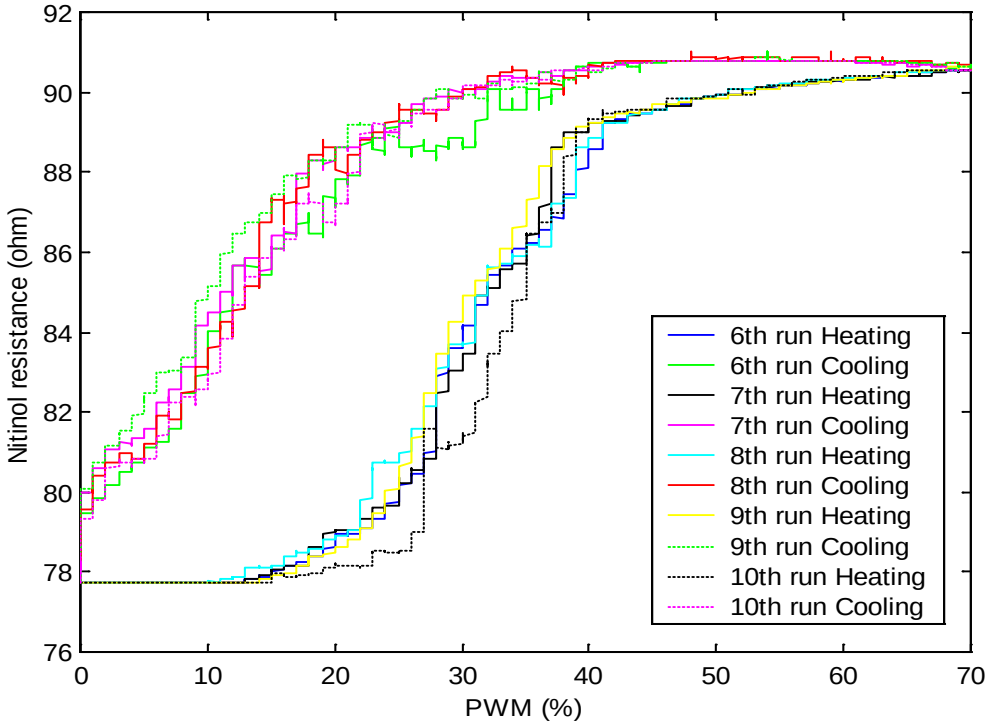


Figure 4-20 The graph of wire resistance –PWM characteristics of the SMA actuator (runs 6 – 10).

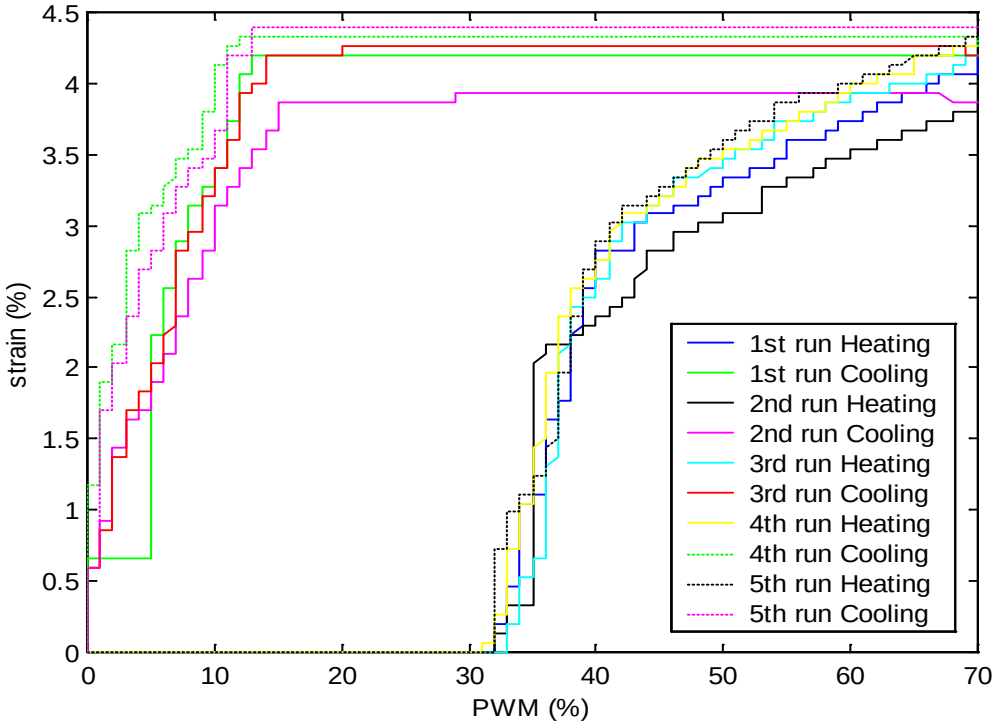


Figure 4-21 The graph of strain –PWM characteristics of the SMA actuator (runs 1 – 5).

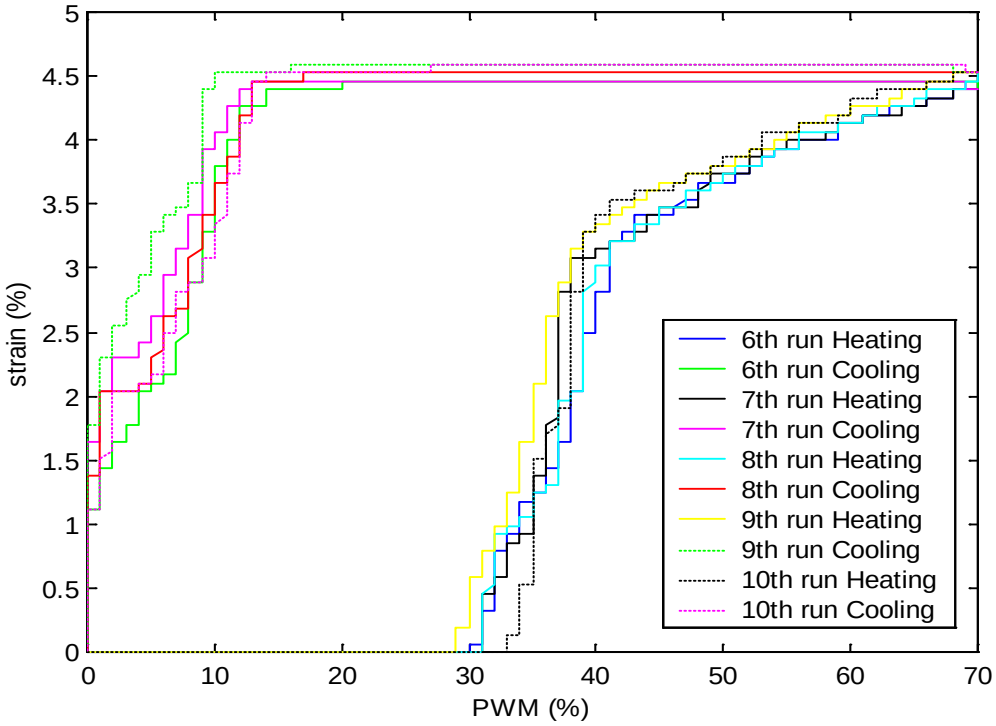


Figure 4-22 The graph of strain –PWM characteristics of the SMA actuator (runs 6 – 10).

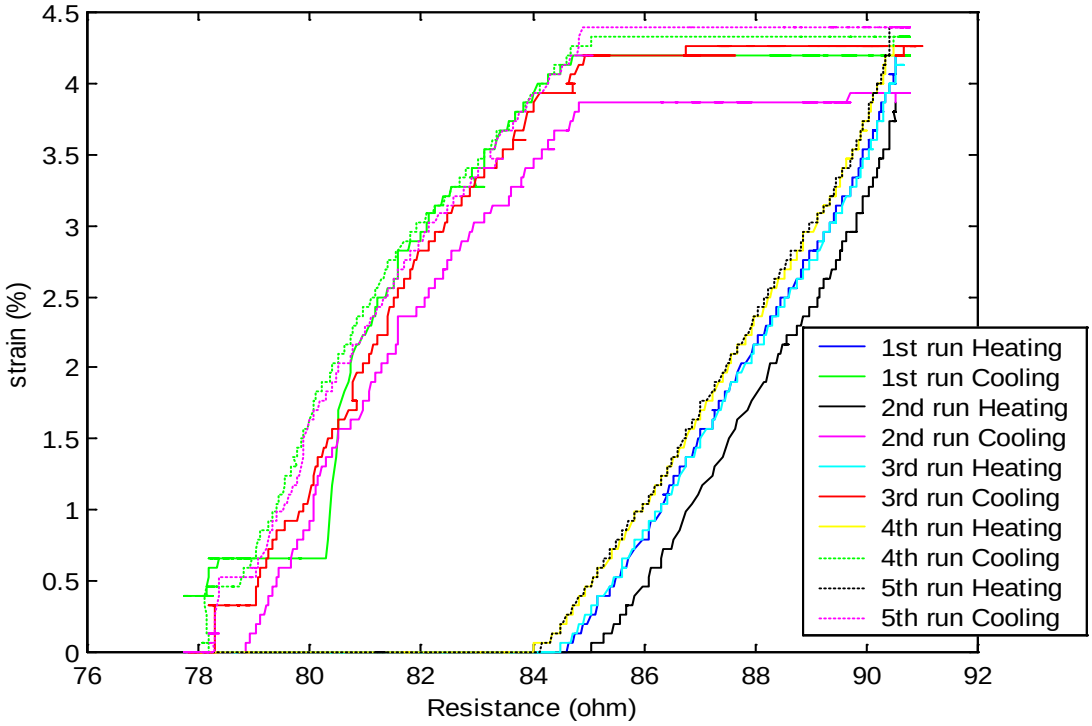


Figure 4-23 The graph of resistance – strain characteristics of the SMA actuator (runs 1 – 5).

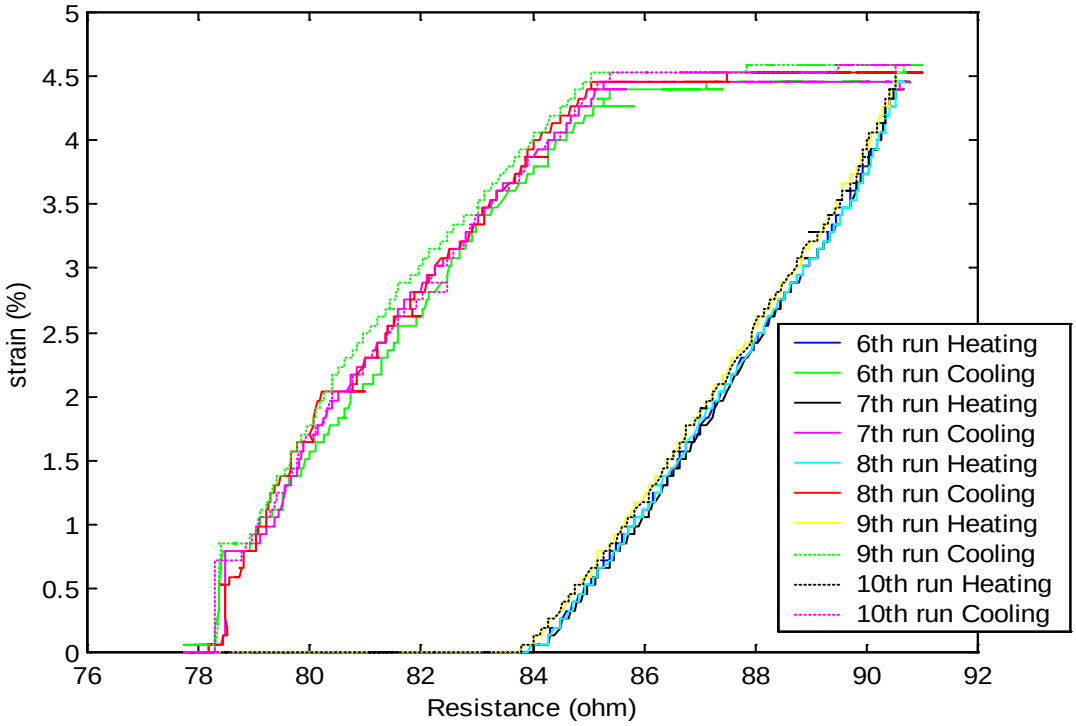


Figure 4-24 The graph of resistance – strain characteristics of the SMA actuator (runs 6 – 10).

Figs 4.25 to 4.27 show the results of a minor hysteresis nested within the main hysteresis. The results were obtained by heating the actuator using maximum percentage of PWM then lowered to 20% PWM, next the PWM was adjusted back to maximum again. This action would avoid the transition state (when both martensite and austenite coexist) from transforming to fully martensite. The nitinol wires was then cooled back to ambient temperature by zeroing the PWM. The shifting phenomenon introduced a challenging problem in maintaining the PWM percentage to the point just before the transformation temperature start.

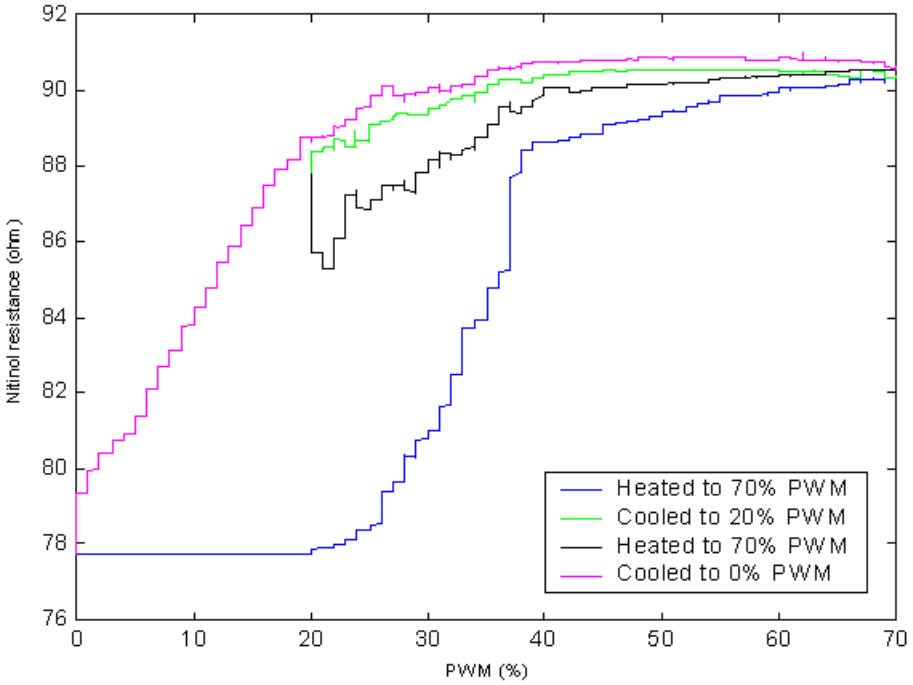


Figure 4-25 The graph of minor hysteresis loop with wire resistance – PWM characteristics of the SMA actuator.

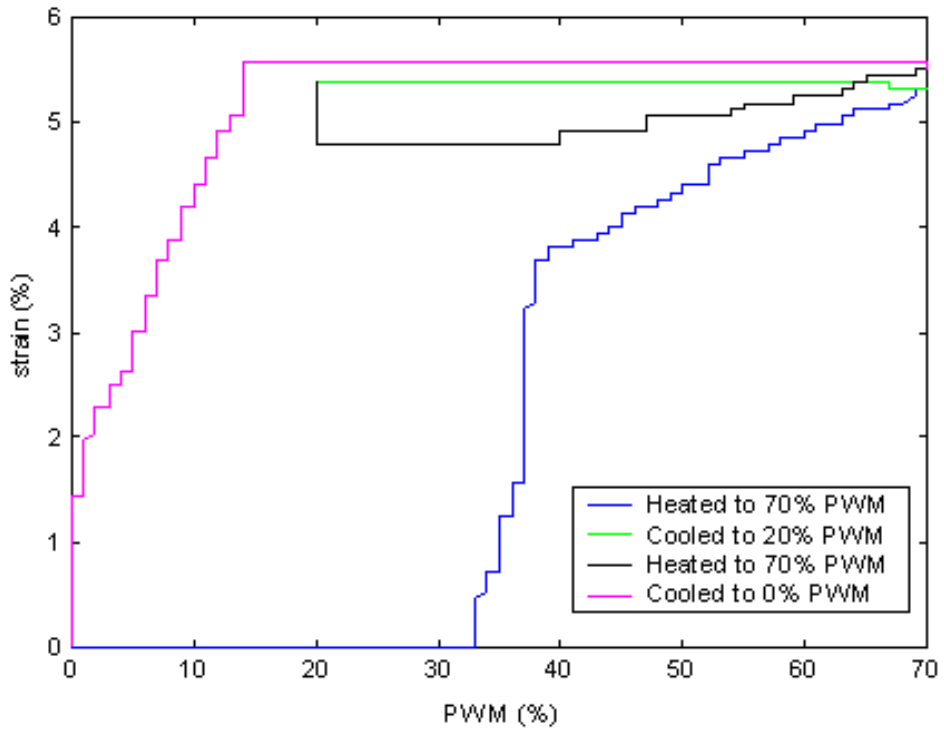


Figure 4-26 The graph of minor hysteresis loop with wire strain – PWM characteristics of the SMA actuator.

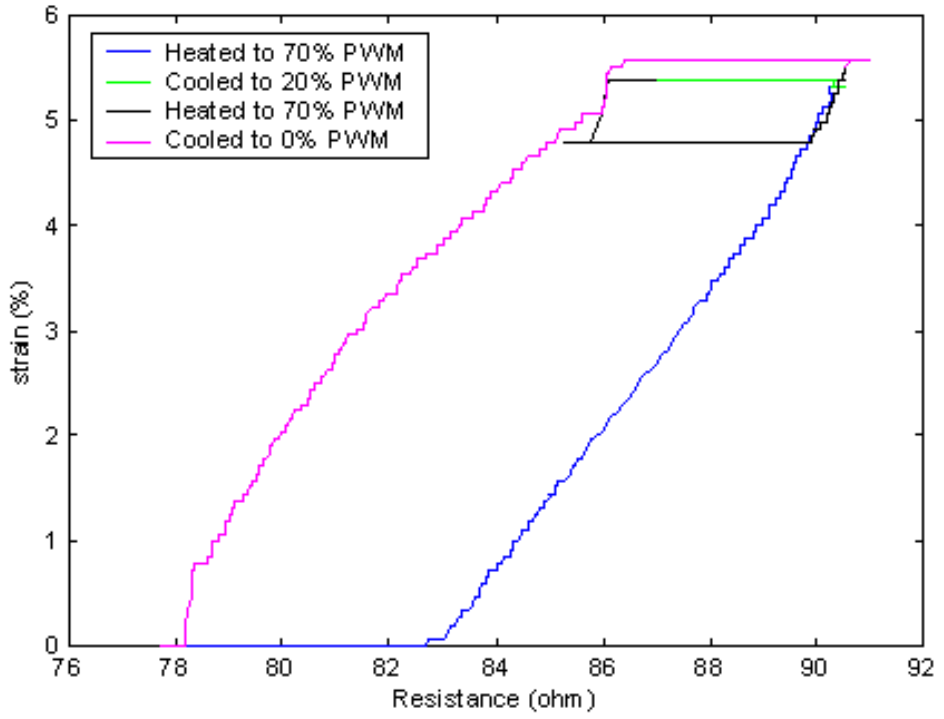


Figure 4-27 The graph of minor hysteresis loop with wire resistance – strain characteristics of the SMA actuator.

4.5 Cyclic Behaviour Training

In general, SMA actuators must operate with repeated cyclic motion. Therefore, the investigation of the gradual decrease in recovery force with cyclic motions as well as measures to prevent the degradation in shape memory effect are very important for the actual use of SMA as a robot actuator. Based on the experimental results, Tobushi [15] concluded that the rate of reduction of recoverable elongation can be slowed down by introducing training before it is employed in any applications. However the specimens that Tobushi [15] tested was a SMA coil. It is necessary to investigate the phenomenon for a SMA wire, so simular tests were carried out on SMA wire.

The apparatus was almost exactly the same as it was when testing with the use of resistance monitoring circuit (refer section 4.2), except the load weight was replaced with a low stiffness spring. By varying a low stiffness spring, the SMA wire eventually straightens without introducing too much stress. Once the apparatus has been set up, the SMA wire was then heated up and cooled down by PWM current control for a thousand hysteresis cycles. At the time when this SMA training was investigated, the actual SMA mechanism was not yet built. Therefore it is necessary to test the SMA actuator training in a simular mechanism, in order to determine the difference between both trained and untrained SMA actuators in terms of irrecoverable elongation Δl_p. Thus the closest model would be using constant dead weight rather than a spring as was used in these measurements.

The relation between the irrecoverable elongation of the SMA Δl_p and the number of cycles N is shown in Fig 4.28. The irrecoverable elongation Δl_p is represented by

$$\Delta l_p = (L - L_o)/L_o \qquad (4.1)$$

where L denotes the axial length of the SMA wire under 70% PWM and L_o denotes the shape-memorized initial length. In Fig 4.28, the blue curve is the result for the cases subject to the training of $N = 10^3$ and show the relation between Δl_p, which was calculated by using L at $N = 10^3$ as L_o in eqn 4.1, and the number of cycles ($N \times 10^3$).

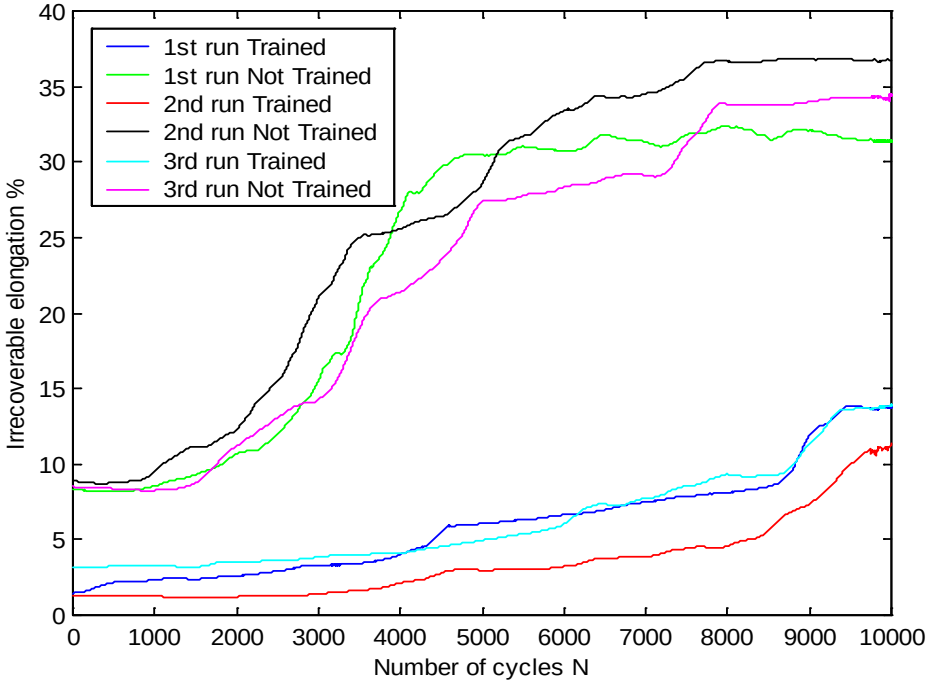


Figure 4-28 Changes in irrecoverable elongation as a function of the repletion cycles.

As observed in Fig. 4.28, Δl_p increases with an increase in N. In the worse scenario 37% of the 5% strain was not recoverable. If the SMA wire is subjected to the training, Δl_p becomes small. After 10^5 cycles of testing, the tremendous difference of Δl_p is caused by the fact of training. Thus it may be said that, if the SMA wire subject to the training is used in an application, the appearance of the irrecoverable elongation can be avoided.

4.6 Conclusion

Several testing methods were used to gain understanding of the characteristics of the SMA actuator as well as to verify the results from the literature. Tests results show that the average hysteresis happens between 15% to 40% PWM signal for 18V applied to the actuators design and these provide up to 5% strain. The hysteresis produces up to 10Ω difference in resistance from 76Ω to 87Ω. The results are repeatable so the wire showed promise as a useful actuator that can use electrical resistance measurements to determine the SMA state for feedback control.

Chapter 5 Orientation of the Stewart Platform

5.1 Formulation for the changes in position and orientation

It is often useful to determine the changes in position and orientation of the Stewart Platform, in order to monitor the motion of the end-effector. For this reason a general and approximate method is presented in the following. This approximated method is applicable to small displacement and it is computationally efficient, and its implementation is straightforward.

Consider a 6 dof symmetric Stewart Platform as shown in Fig 5.1. Points p_i (i=1,2,3,4,5,6) are the six non-collinear points of the moving platform under consideration where each points p_i is reference to the end-effector reference frame $\{O_e -X_e -Y_e -Z_e\}$. Where the coordinates of the six points expressed with respected to the base reference frame {O-X-Y-Z} are labelled as G_i (i=1,2,3,4,5,6).

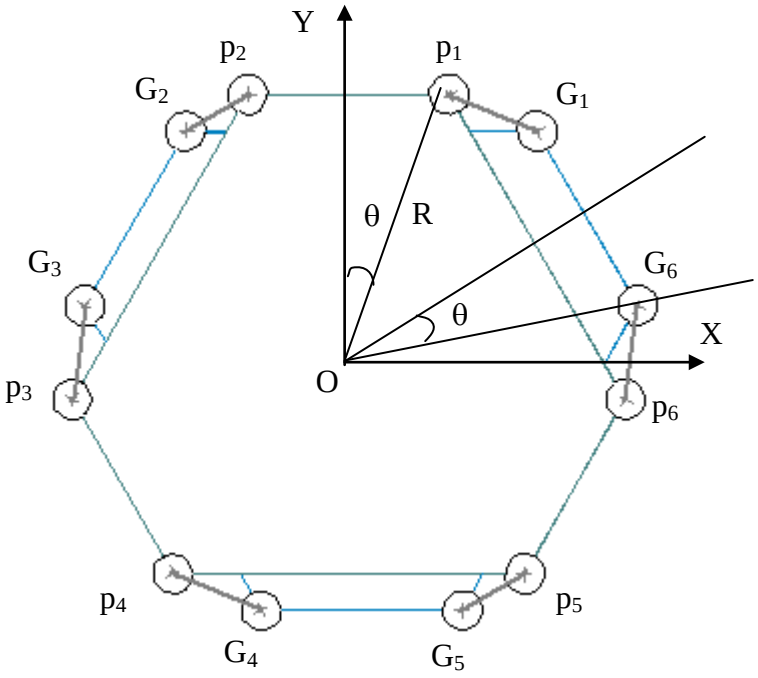


Figure 5-1 The top view of a symmetric 6 DOF platform.

For a symmetric Stewart Platform points p_i and G_i would lie in a same radius R. (Refer to Fig. 5.1). Thus each point p_i can be represented as follows

$$\underline{p}_1 = R\begin{bmatrix}\cos(90^\circ-\theta)\\ \sin(90^\circ-\theta)\\ 0\end{bmatrix},\ \underline{p}_2 = R\begin{bmatrix}\cos(90^\circ+\theta)\\ \sin(90^\circ+\theta)\\ 0\end{bmatrix},$$

$$\underline{p}_3 = R\begin{bmatrix}\cos(210^\circ-\theta)\\ \sin(210^\circ-\theta)\\ 0\end{bmatrix},\ \underline{p}_4 = R\begin{bmatrix}\cos(210^\circ+\theta)\\ \sin(210^\circ+\theta)\\ 0\end{bmatrix},$$

$$\underline{p}_5 = R\begin{bmatrix}\cos(330^\circ-\theta)\\ \sin(330^\circ-\theta)\\ 0\end{bmatrix} \text{ and } \underline{p}_6 = R\begin{bmatrix}\cos(330^\circ+\theta)\\ \sin(330^\circ+\theta)\\ 0\end{bmatrix}$$

Whereas points G_i reference to the base reference frame {O-X-Y-Z} can be represented by

$$\underline{G}_1 = R\begin{bmatrix}\cos(30^\circ+\theta)\\ \sin(30^\circ+\theta)\\ 0\end{bmatrix},\ \underline{G}_2 = R\begin{bmatrix}\cos(150^\circ-\theta)\\ \sin(150^\circ-\theta)\\ 0\end{bmatrix},$$

$$\underline{G}_3 = R\begin{bmatrix}\cos(150^\circ+\theta)\\ \sin(150^\circ+\theta)\\ 0\end{bmatrix},\underline{G}_4 = R\begin{bmatrix}\cos(270^\circ-\theta)\\ \sin(270^\circ-\theta)\\ 0\end{bmatrix},$$

$$\underline{G}_5 = R\begin{bmatrix}\cos(270^\circ+\theta)\\ \sin(270^\circ+\theta)\\ 0\end{bmatrix} \text{ and } \underline{G}_6 = R\begin{bmatrix}\cos(30^\circ-\theta)\\ \sin(30^\circ-\theta)\\ 0\end{bmatrix}$$

Where vector pointing from the origin of the base frame to the origin of the end-effector frame is

$${}^{G}_{p}\underline{P} = \underline{P}_o = \begin{bmatrix}0\\ 0\\ 83.5\end{bmatrix}$$

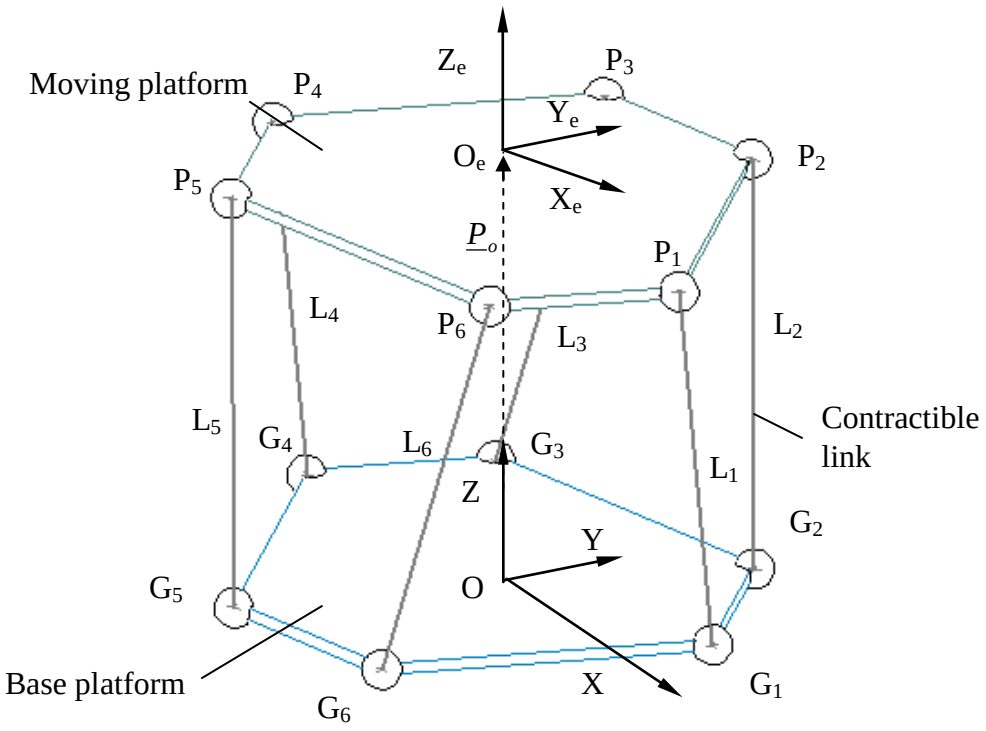


Figure 5-2 A general 6 DOF Stewart Platform.

Consider the orientation γ about x, β about y and α about z of platform. The rotation matrix can be constructed as follow:

$$ {}^{G}_{p}R_{xyz}(\gamma,\beta,\alpha) = R_z(\alpha)R_y(\beta)R_x(\gamma) $$

$$ = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} $$

Since small perturbation: $\varepsilon \approx \beta \approx \gamma << 1\,\text{rad}$, the rotation matrix can be further simplified to:

$$ {}^{0}_{0}R_{xyz}(\gamma,\beta,\alpha) = \begin{bmatrix} 1 & -\alpha & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & 0 \\ -\beta & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\gamma \\ 0 & \gamma & 1 \end{bmatrix} \qquad \beta\gamma \approx 0 $$

$$ = \begin{bmatrix} 1 & -\alpha & \beta \\ \alpha & 1 & -\gamma \\ -\beta & \gamma & 1 \end{bmatrix} $$

This is the first order approximation.

So for the 6 lengths of the actuators (i = 1,2,3,4,5,6) are expressed as follow. The equation is driven by six variables, which are Δx, Δy, Δz, Δα, Δβ and Δγ.

$$\underline{L}_i = \underline{P}_{(xyz)} - \underline{G}_i + {}^{G}_{p}R_{xyz}(\gamma,\beta,\alpha)\,{}^{P}\underline{p}_i \qquad (5.1)$$

Where the initial lengths $\underline{L}_{io}$ of the actuators with no change in all axis and all rotation angles are expressed as follow.

$$\underline{L}_{io} = \underline{P}_o - \underline{G}_i + {}^{G}_{p}R_{xyz}(0,0,0)\underline{p}_i = \underline{P}_o - \underline{G}_i + \underline{p}_i \qquad (5.2)$$

With eqn.5.1 and 5.2 the net change in length can be expressed as follow:

$$\underline{\Delta L}_i = \underline{L}_i - \underline{L}_{io} = \underline{P}_{(xyz)} - \underline{P}_o + \begin{bmatrix} 0 & -\alpha & \beta \\ \alpha & 0 & -\gamma \\ -\beta & \gamma & 0 \end{bmatrix} {}^{p}\underline{p}_i \qquad (5.3)$$

Define the whole set of 64 possible orientation of the Stewart Platform by a binary no. B={B1, B2, B3, B4, B5, B6} where Bi = 0, 1. Thus the expression for ΔLi can be rewritten (see eqn. 5.4). Where K is a constant.

$\Delta L_i = KB_i$ and $\underline{P}_{(xyz)} = \underline{P}_B$ (5.4)

Where $\underline{P}_{(xyz)} - \underline{P}_o = \underline{\mathbb{P}}_B$

Equation 5.3 can be further simplified to:

$$\underline{\Delta L}_i = \underline{\mathbb{P}}_B + \mathbb{R}\,{}^{p}\underline{P}_i \qquad (5.5)$$

and $\left|\underline{\Delta L}_i\right| = \Delta L_i = KB_i$

The contraction is that since ΔLi<<Li, then we can approximate the changes in position and orientation of the platform from P by 1st order effects.

$$\underline{\mathbb{P}}_B = \sum_{i=1}^{6} \underline{\mathbb{P}}\,(2^{i-1}) \qquad (5.6)$$

With eqn 5.3 the change in position and orientation of the Stewart Platform can be related with a Jacobian matrix, which can be written as below.

$$\Delta \underline{L} = \underline{J} \Delta \underline{Q}$$

$$\text{or } \Delta \underline{Q} = \underline{J}^{-1} \Delta \underline{L} \qquad (5.7)$$

With six actuators and six independent variables the size of the Jacobian matrix is 6 x 6. The first three columns of the Jacobian matrix can be derived by varying Δx, Δy and Δz one at a time and hold Δα, Δβ and Δγ constant. Thus which give:

$$\Delta L_i = \hat{\underline{L}}_i \cdot \Delta \underline{L}_i$$

$$\frac{\Delta L_i}{\Delta x} = \frac{L_{ix}}{|L_i|} \;,\; \frac{\Delta L_i}{\Delta y} = \frac{L_{iy}}{|L_i|} \;,\; \frac{\Delta L_i}{\Delta z} = \frac{L_{iz}}{|L_i|} \qquad (5.8)$$

Whereby the last three columns can be derived by varying Δα, Δβ and Δγ one at a time and hold Δx, Δy and Δz constant. Thus which give:

$$\Delta \underline{L_o} = \begin{bmatrix} 0 & -\Delta\alpha & 0 \\ \Delta\alpha & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \left(P_x \hat{i} - P_y \hat{j}\right) \Delta\alpha$$

$$\Delta \underline{L_i} = \Delta \underline{L_i} \cdot \hat{\underline{L}}_i = \frac{p_{ix} L_{ix} - p_{iy} L_{iy}}{|L_i|} \Delta\alpha$$

$$\frac{\Delta L_o}{\Delta \alpha} = \frac{p_{ix} L_{ix} - p_{iy} L_{iy}}{|L_i|} \qquad (5.9)$$

Similarly $\quad \dfrac{\Delta L_o}{\Delta \beta} = \dfrac{p_{iz} L_{iz} - p_{ix} L_{ix}}{|L_i|}, \dfrac{\Delta L_o}{\Delta \gamma} = \dfrac{p_{iy} L_{iy} - p_{iz} L_{iz}}{|L_i|}$

By substitute eqn 5.8 and 5.9 to eqn 5.7.

$$\begin{bmatrix} \Delta L_1 \\ \vdots \\ \Delta L_N \end{bmatrix} = \begin{bmatrix} \frac{L_{1x}}{|L_1|} & \frac{L_{1y}}{|L_1|} & \frac{L_{1z}}{|L_1|} & \frac{p_{1x}L_{1x} - p_{1y}L_{1y}}{|L_1|} & \frac{p_{1z}L_{1z} - p_{1x}L_{1x}}{|L_1|} & \frac{p_{1y}L_{1y} - p_{1z}L_{1z}}{|L_1|} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{L_{Nx}}{|L_N|} & \frac{L_{Ny}}{|L_N|} & \frac{L_{Nz}}{|L_N|} & \frac{p_{Nx}L_{Nx} - p_{Ny}L_{Ny}}{|L_N|} & \frac{p_{Nz}L_{Nz} - p_{Nx}L_{Nx}}{|L_N|} & \frac{p_{Ny}L_{Ny} - p_{Nz}L_{Nz}}{|L_N|} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \alpha \\ \Delta \beta \\ \gamma \end{bmatrix}$$

Where N=6.

5.2 Simulated results

To illustrate the application of the method discussed above, let us consider a numerical example for a 6 dof symmetric Stewart Platform shown in Fig 5.1.The moving and base platforms of the manipulator have the same dimensions with R = 41.169 mm and the angle θ is 13.71°. Vectors p_i and G_i are expressed as follows:

$$\underline{p}_1 = \begin{bmatrix} 9.757 \\ 39.996 \\ 0 \end{bmatrix}, \ \underline{p}_2 = \begin{bmatrix} -9.757 \\ 39.996 \\ 0 \end{bmatrix}, \ \underline{p}_3 = \begin{bmatrix} -39.5167 \\ -11.547 \\ 0 \end{bmatrix},$$

$$\underline{p}_4 = \begin{bmatrix} -29.7583 \\ -28.4489 \\ 0 \end{bmatrix}, \ \underline{p}_5 = \begin{bmatrix} 29.7583 \\ -28.4489 \\ 0 \end{bmatrix}, \ \underline{p}_6 = \begin{bmatrix} 39.5167 \\ -11.547 \\ 0 \end{bmatrix}$$

$$\underline{G}_1 = \begin{bmatrix} -29.7583 \\ 28.4489 \\ 0 \end{bmatrix}, \ \underline{G}_2 = \begin{bmatrix} -29.7583 \\ 28.4489 \\ 0 \end{bmatrix}, \ \underline{G}_3 = \begin{bmatrix} -39.5167 \\ 11.547 \\ 0 \end{bmatrix},$$

$$\underline{G}_4 = \begin{bmatrix} -9.7583 \\ -39.9959 \\ 0 \end{bmatrix}, \ \underline{G}_5 = \begin{bmatrix} 9.7583 \\ -39.9959 \\ 0 \end{bmatrix}, \ \underline{G}_5 = \begin{bmatrix} 39.5167 \\ 11.547 \\ 0 \end{bmatrix}$$

A small displacement of 0.001 mm and rotation of 0.1 rad are taken as an example. Thus the resultant length of Li (I=1,2,3,4,5,6) can be calculated by using eqn 5.1 and by varying each parameter one at a time the results are summarised as follows.

Table 5.1 Summarised results for Li

	Initial state	Change in x-axis	Change in y-axis	Change in z-axis	Rotate about z-axis	Rotate about y-axis	Rotate about x-axis
Magnitude of L_1 (mm)	86.6348	86.6345	86.6349	86.6357	87.7784	85.6946	90.4959
L_2	86.6348	86.635	86.6349	86.6357	85.6739	87.5757	90.4959
L_3	86.6348	86.6348	86.6345	86.6357	87.7784	90.4496	85.5224
L_4	86.6348	86.6345	86.6349	86.6357	85.6739	89.5064	83.8962
L_5	86.6348	86.635	86.6349	86.6357	87.7784	83.7704	83.8962
L_6	86.6348	86.6348	86.6345	86.6357	85.6739	82.8328	85.5224

The results are within expectations especially when there are a change along z-axis and rotation about the z-axis. As it can be seen that when there is a change along z-axis, the length of all the actuators are same. The change in z-axis is simular to either bringing the end-effector closer to the base or to bring it further away from the base. Both of these actions would require all the actuators contract/extend together. This result to all the actuators' length would be the same. The next case is when there is a rotation about the z-axis, two groups of the three actuators' length are equal (refer to Fig 5.3).

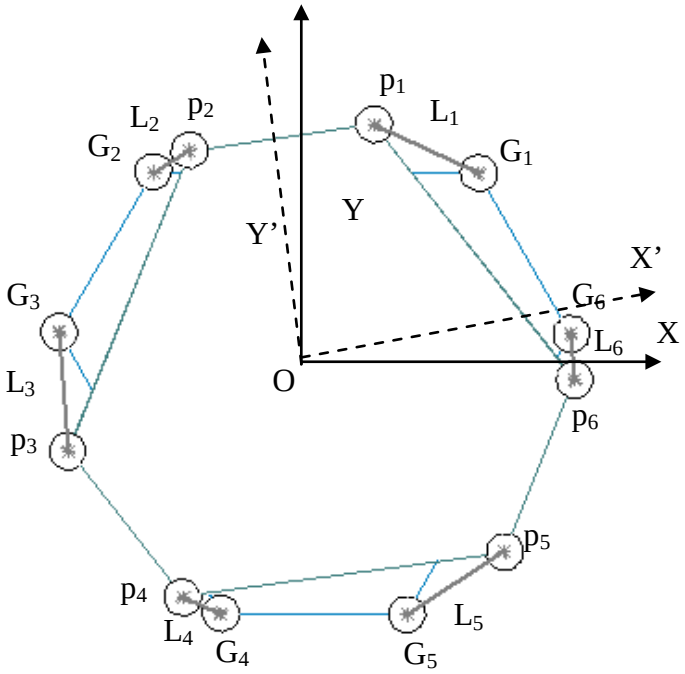


Figure 5-3 The top view of a symmetric 6 DOF platform rotated about z axis.

From then the Jacobian matrix can be formed by using eqn 5.8 and 5.9 as follows

$$J = \begin{bmatrix} -0.2308 & 0.1333 & 0.9638 & -7.5835 & -2.2234 & 5.3893 \\ 0.2309 & 0.1333 & 0.9638 & -7.5835 & -2.2780 & 5.2735 \\ 0.0000 & -0.2666 & 0.9638 & -3.0780 & 0.0000 & 2.9482 \\ -0.2308 & 0.1333 & 0.9638 & 10.6615 & 6.9469 & -3.6701 \\ 0.0000 & 0.1333 & 0.9638 & 10.6615 & 6.7803 & -3.9214 \\ 0.0000 & -0.2666 & 0.9638 & -3.0780 & 0.0000 & 3.2193 \end{bmatrix}$$

5.3 Conclusion

A method is proposed based on the points position to approximate the change in position and orientation of the platform from P by 1st order effects. The method is general, easy to implement, and can be used for any 6 dof platform type parallel mechanism. A symmetrical 6 dof Stewart Platform is discussed in detail as an example to illustrate the solution procedures of the methods.

Chapter 6 Hardware Implementation and Synergy of Control System

6.1 General view of the design of the circuit board

An overview of the electrical arrangement for a Nitinol actuated four section Stewart Platform. Dr Dunlop suggested that the electrical control of the Stewart Platform be based on two printed circuit boards (PCB) located within every second deck of the four stages Stewart Platform robot. These circuit boards were trimmed to a hexagonal shape fitted firmly into the platform deck. The circuit board included three sets of eight female connectors at three sides as a means of connecting the male connectors from the nitinol wires to the circuit board. Each set of female connector was connected up to four nitinol actuated leaf springs. Communications from deck to deck can be accomplished through ribbon cables. A simple schematic view is shown below.

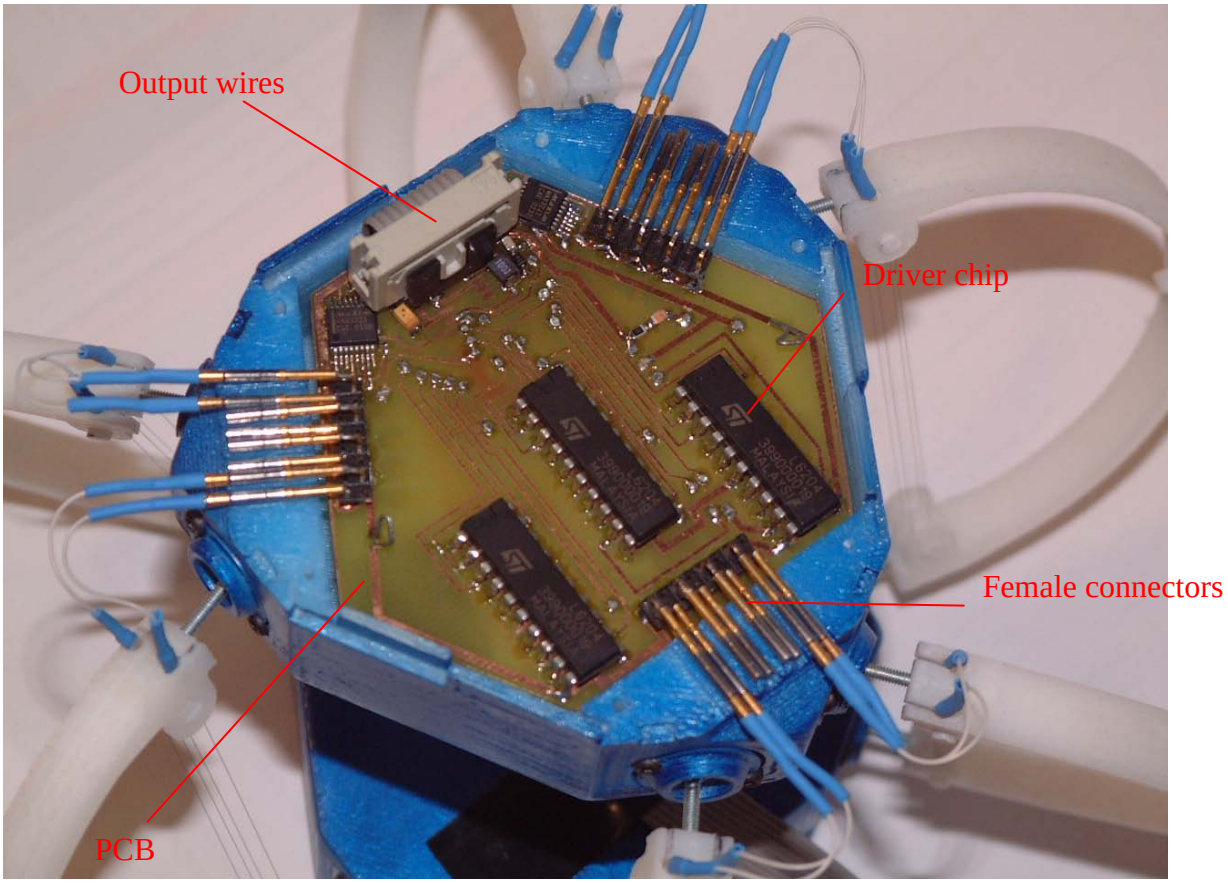


Figure 6-1 Main printed circuit board.

The size of the circuit board was dependent by the surrounding geometry, and the thickness circuit is 1.5mm. The PCB was held in place by the three female connector blocks at three edges of the platform deck.

Each circuit board was controlled by a single microchip; the 28 lead PSoC (Programmable System on a Chip) from Cyprus Microsystems. The PSoC microchip is versatile as it generates PWM (Pulse Width Modulation) for the nitinol wire, and also contains the A/D conversion, programmable gain amplifier and multiplexer for measuring the differential voltages for the 3 driver chips. See figure below.

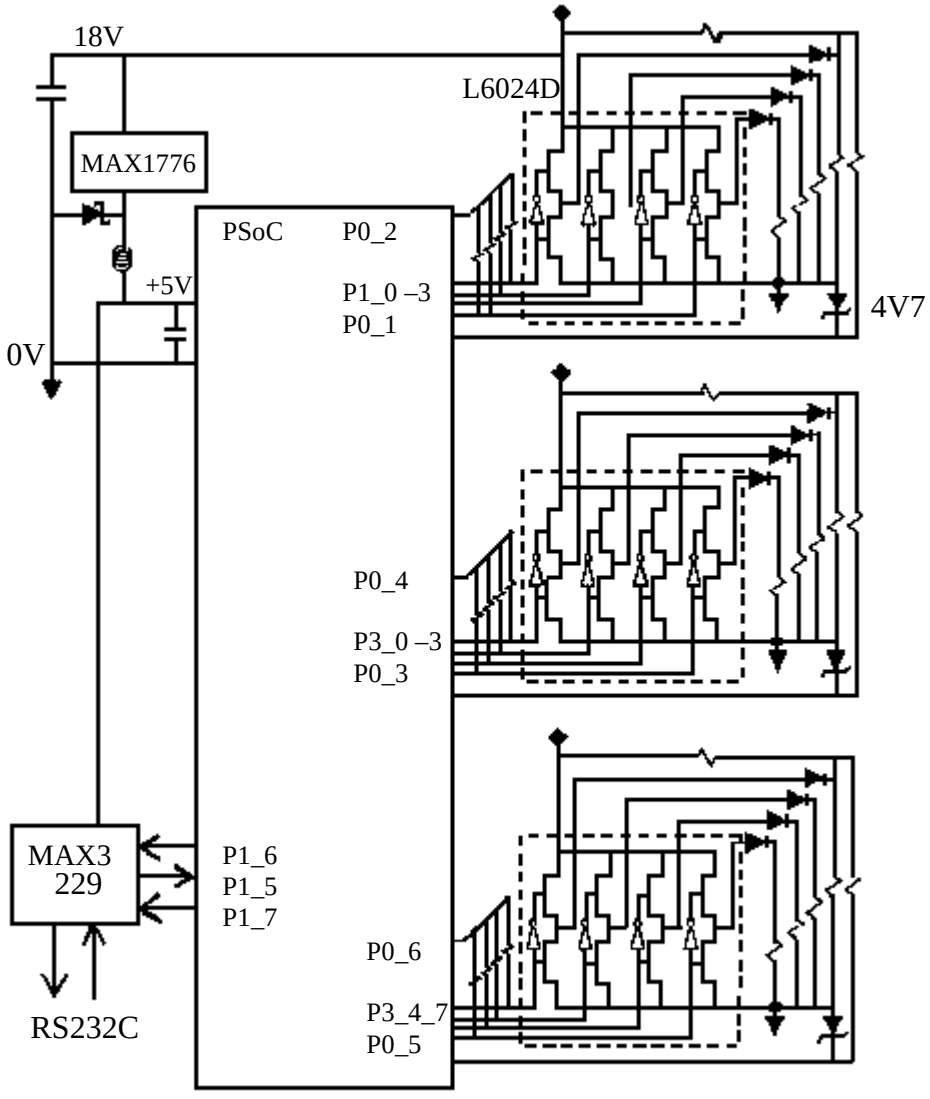


Figure 6-2 A schematic view of the circuit board design.

The 18V voltage supply for this PCB drives the driver chips and the PSoC. However the PSoC requires 5V which is provided by the MAX1776 switching regulator, inductor and schottky diode.

In each block of four actuators one of the nitinol wires is monitored to determine its resistance during the off portion of the PWM. During the on portion of the PWM drive, 18V is applied to the nitinol wire element via a series diode. Thus 18V can be applied to the analog input (P0_1, P0_3 or P0_5) of the PSoC. The series resistor and 4.7V zener prevent this from happening.
To accomplish the communication between the PSoC and a PC, a serial RS232 link was employed. The serial port from a PC needs the serial transmission to be converted from ±12V to 0-5V which is accomplished by the MAX3229.

6.2 Hardware detail

The nitinol actuator is driven by a state machine having 4 states. Thus the driver chip operates in 4 states: on, off, PWM, and off. The transitions are determined by measurements of the electrical resistance of the nitinol wire is shown in Fig 2.4. The idea is to switch on the electric current until the Nitinol wire heats to above 80ºC and then the current is switched off until the wire cools to around 30ºC while still maintaining the same strain. PWM is then used to maintain the temperature at the knee of the hysteretic strain cycle. When the current is switched off, the wire cools and the strain returns to its original state. The resistance measurement system operates whenever the heating switch is off. Once the switching transient has died away, the differential measurement provides a voltage proportional to the electrical resistance of the Nitinol wire, and proportional to the change of strain. A schematic view of the driving chip is shown in Fig. 6.3, and a simplified version is shown in Fig 6.4.

The flow of the current to the nitinol wire i can be determined just by the control of bit B_i or alternatively by setting the bit B_i into tristate to let the PWM signal determine the drive. This allows the PSoC microchip to send a PWM signal to a SMA actuator (provided the corresponding bit is set to tristate) or to switch it full, off or on. The advantage of this approach is to keep the workload of the PSoC microchip as low as possible. Note when bit $B_i = 1$ the corresponding top FET would set to on else $B_i = 0$ the corresponding top FET would be switched off. $B_i = 0$ also turns on the lower FET but the series diode prevents this FET having any effect.

The driver system was designed bases on the experimental results collected (see chapter 4.4). The range of the resistance hysteresis lies between 76Ω to 90Ω. However the range of the hysteresis is subjected to change under the variation of environment conditions, such as applied load and ambient temperature. It is important to ensure that the circuitry could adapt to variation of the hysteresis. Thus the design range has to be wide enough to cover every nitinol hysteresis loop. For this reason each actuator was tested individually (See Appendix D). The design range was consequently set to be from 65Ω to 100Ω.

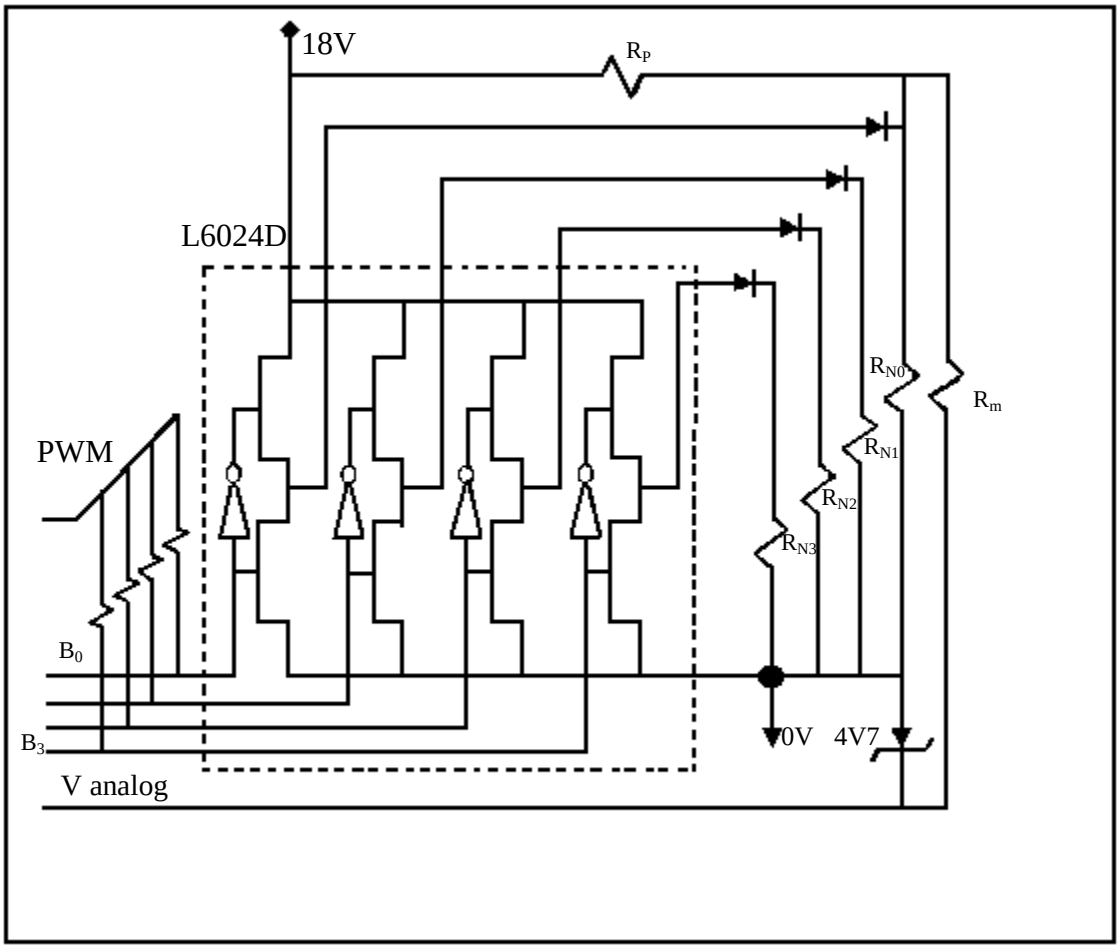


Figure 6-3 A basic layout of the driver chip.

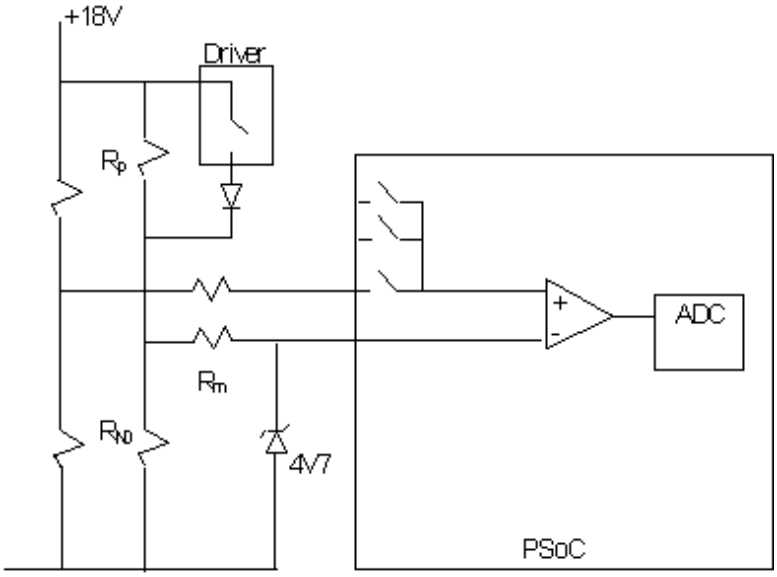


Figure 6-4 Simplified schematic of Wheatstone bridge (when driver is off)

In Fig. 6.3 the nitinol wire that actuates the leaf spring is represented as R_{Ni} I=0, 4or 8. During the off period of the PWM, R_p supplies voltage tn $R_{N0.}$ The purpose of this is to enable the measurement of the voltage R_{N0} to find the resistance, but without generating too much heat from in the nitinol wire. Around 1V is used for this purpose and it provides less than $(1/18)^2$ or 0.3% of full power for heating.

Thus the calculations for designing the size of the resistor R_p is carried out as follows,
for 1V

$$I = \frac{V_N}{R_N} = \frac{1V}{65\Omega} = 0.0153A \text{ max}$$

$$R_p = \frac{V_p}{I} = \frac{17}{0.0153} = 1105\Omega$$

R_p is expected to be about ≈1100Ω.

If the nitinol resistance is between 65Ω to 100Ω and the supply voltage is 18V. Then

$$V_{\min} = \frac{18V}{1165\Omega} \times 65\Omega = 1.004V$$

$$V_{\max=} \frac{18V}{1200\Omega} \times 100\Omega = 1.5V$$

$$\Delta V = 0.496\text{V}$$

In order to provide the maximum voltage range out of the programmable gain differential amplifier, the analog ground is set to a mean value of 1.25V. Thus a differential voltage of ±0.25V was obtained for input to the amplifier with a gain of 4, so that the voltage is matched to the ADC. As a result, resistor R_p is chosen to be 1.1kΩ.

Although the electronic circuit board was still a “work in progress” in the electronics workshop when this thesis was prepared, it was possible to carry out some static tests by adjusting the voltage on the nitinol actuator so as to achieve the correct operating temperatures. This is described in the next chapter.

6.3 Alternative control system

SMA has temperature hysteresis, which severely degrades system performance such as giving rise to undesirable inaccuracy. The temperature hysteresis width between austenite phase and martensite phase is about 40°C. To control the displacement of SMA actuators precisely, such hysteresis must be compensated. In order to compensate hysteresis, the control system used two reverse models, where this modelling method is developed by Hasegawa [16]. The inverse model which locates in front of the SMA actuator can cancel its hysteresis completely, if the inverse model represents the whole behaviour of the actuator. The static hysteresis model, however, cannot represent the dynamic behaviour of actuators. In order to compensate the hysteresis in a dynamic plant, it divides into a dynamic element and static hysteresis, it considers, and it proposes the control system which controls each by using two reverse models separately. The electrical resistance of SMA depends on the process of phase transformation, so the control of it is essential for controlling SMA actuators.

6.4 Hysteresis Model

The model developed by Hasegawa [16] represents a relationship between a thermal input u(k) and the fraction of martensite y(k). Suppose the model is assumed to contain N elementary relays whose output is switched from one to zero at the input α_i [I: I = 1~N] and switched from one to zero at the input β_i. Every weighting function is set to 1/N and it is estimated that the distribution of switching points α_i and β_i, instead of estimating $\mu(\alpha,\beta)$. It is known that increasing the stress is proportional to the axial load. Thus, α_i and β_i are described in terms of the axial load P and constant coefficients C_α and C_β. Then the model is expressed as follows:

$$y(k) = Hys(u(k)) = \frac{1}{N}\sum_{i=1}^{N} v_{\alpha_i,\beta_i}(u(k)) \qquad (6.1)$$

where

$$\alpha_i = \alpha_{i0} + C_\alpha P$$
$$\beta_i = \beta_{i0} + C_\beta P$$

α_{i0} and β_{i0} are transformation temperatures before loading, and $A_s \le \alpha_{i0} \le A_f$, $M_f \le \alpha_{i0} \le M_s$. The parameters α_{i0} and β_{i0} are determined as

$$\alpha_{i0} = \frac{A_s + A_f}{2} + nrd\left(A_f - A_s\right)\cdot 0.35 \qquad (6.2)$$

$$\beta_{i0} = \frac{M_s + M_f}{2} + nrd\left(M_f - M_s\right)\cdot 0.35 \qquad (6.3)$$

where nrd is a normal distributed random number whose mean is zero and whose standard deviation is one. Other parameters used in the simulation are shown in table 6.1.

Table 2.1 Parameters used in simulation

As	60(C
Af	75(C
Ms	45(C
Mf	25(C
C(	0.1N/(C
C(	0.1N/(C
N	500

□ EMBED PBrush □□□

Figure □ STYLEREF 1 \s □6□-□ SEQ Figure * ARABIC \s 1 □5□ Simulation of the hysteresis model

6.5 Inverse Hysteresis

The procedure to obtain the desire at the discrete input temperature u(k) from the desired fraction of austenite ((k) at the kth sampling time is as follows.

1) In the case of $\alpha(k+1) - \alpha(k) \geq 0$

Calculate n1, the number of cells which should be transformed from martensite to austenite.

$n1 = \alpha(k+1) \times N - \alpha(k) \times N$

Transform n1 cells in martensite to austenite in order of α_I magnitude. Then, the output u(k+1) of inverse function, a desired temperature, is given,

$u(k+1) = (\min As_{[i]} \text{ in Martensite} + \max As_{[i]} \text{ in Austenite}) / 2$

2) In the case of $\alpha(k+1) - \alpha(k) \leq 0$

$u(k+1) = (\min Rs_{[i]} \text{ in Austenite} + \max Rs_{[i]} \text{ in Martensite}) / 2$

$R = \alpha \times Ra + (1 - \alpha) \times Rm$

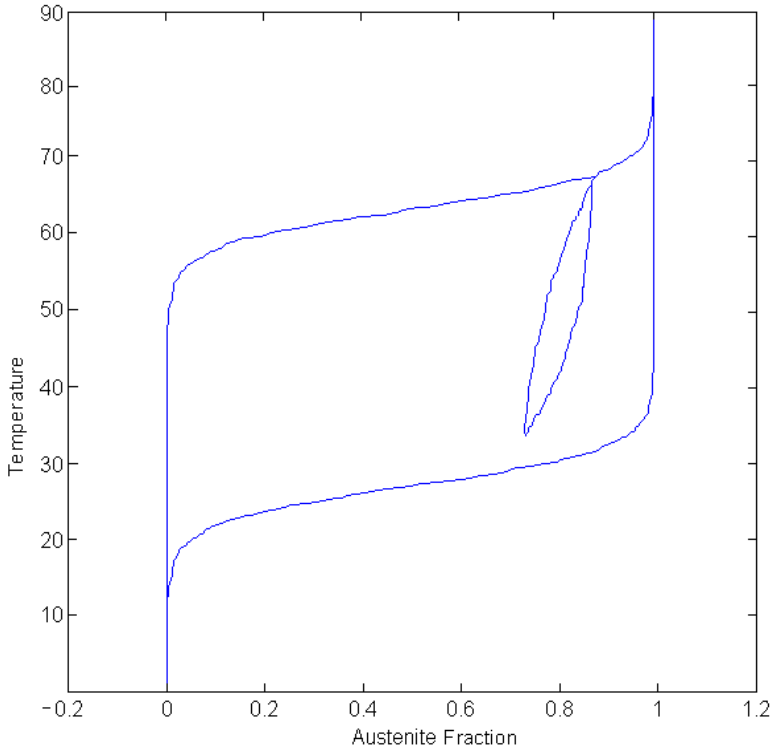


Figure 6-6 Simulation of inverse hysteresis loop

6.6 Plant

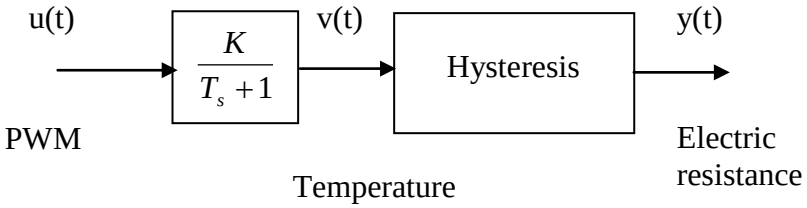


Figure 6-7 Block diagram of SMA actuator [16]

The plant is shown in Fig. 6-7. Where the input of the plant is PWM and the output is the electrical resistance of SMA. The plant contains a first-order lag system $G(s) = K_p/(1+T_ps)$ which translate the PWM to a temperature v(t) and a statical hysteresic property function Hys() in a serial manner. The temperatures of SMA are assumed to be unmeasurable, because the actuator is too small to measure accurately. Here, the input of Hys() is the temperature v(t) and the output is the electrical resistance y(t) which is calculated from α by eqn 6.4.

$$R = \alpha \times Ra + (1 - \alpha) \times Rm \qquad (6.4)$$

Where R_a, R_m and R represent the resistance of SMA in austenite, resistance in martensite and that of complex, respectively.

6.7 Control of electrical resistance

The control system consists of two reverse hysteresis models. See Fig 6.8.

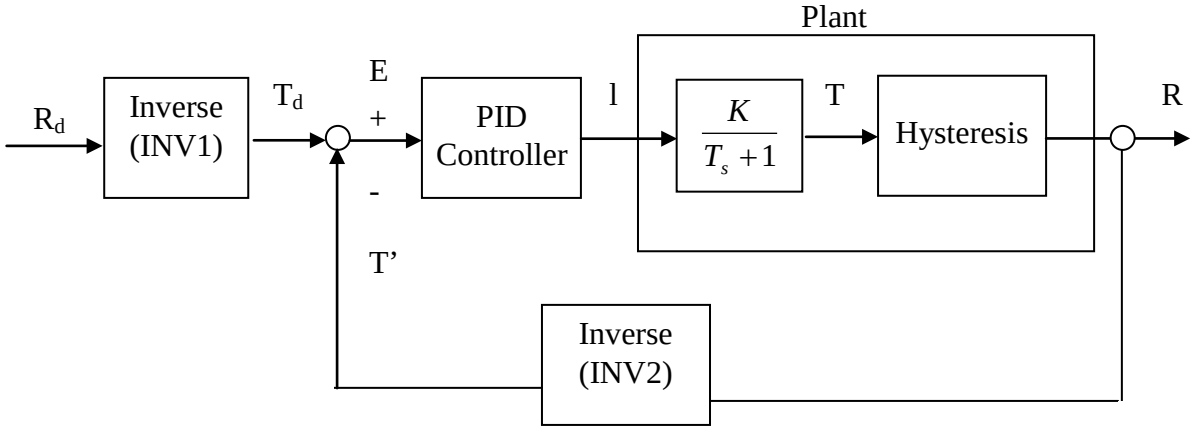


Figure 6-8 Control system equipped with two hysteresis inverse [16]

Although the output of the alignment dynamic system in resistance value and control of the SMA is temperature, it is difficult to measure this correctly using thermocouple. Then it presumes the output of the dynamic system which cannot be measured, using a known reverse hysteresis model as observer. If it takes resistance value control of SMA for an example, using the reverse hysteresis model (INV2) of the hysteretic property between the temperature–resistance value which is known, it presumes temperature of a plant which cannot be measured, and it will feed back this. It changes target resistance value into the target value of temperature by another reverse hysteresis model (INV1) which has the same parameter on the one hand, and it performs PID control so that temperature may become this target value. Even when two reverse hysteresis models and the hysteretic property of the plant are the same parameters, in this control system, the hysteretic property is compensated completely. For example, if it gives a step input and to some extent big proportionality gain in consideration of response characteristic as target value, it will be fallen and attached to the resistance value different from target value. Although a control system moves in order to set temperature to the temperature Td which asked for this by INV1 when it gave Y_{d1} as target value of resistance value, as shown in Fig. 6.9.

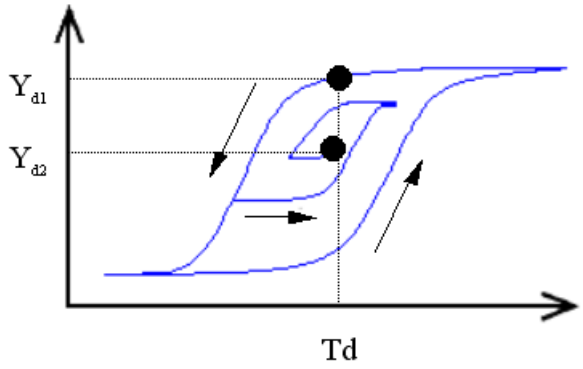


Figure 6-9 Schematic explanation of unreachable value Y_{d1}.

But the feedback loop would introduce an unavoidable minor hysteresis loop. This is because when temperature went too far beyond Td, the control system would attempt to maintain the desire temperature Td by fluctuating the input back and forward. The resistance value will eventually follow the track of the minor hysteresis loop and turn into the value Y_{d2}. Thus differ from Y_{d1}. Then, INV1 was made to calculate target value, in order to improve this with reference to each hysteresis relay element of the reverse hysteresis model (INV2) which inserted a motion of the hysteresis property of a plant in the feedback loop of an output considered to have followed faithfully for every sampling time. By this, a reverse hysteresis model (INV1) can give the target value Td of a suitable temperature every moment, after knowing where it will move. The result which carried out the simulation of the response of a plant to a step input is shown in Fig. 6.10.

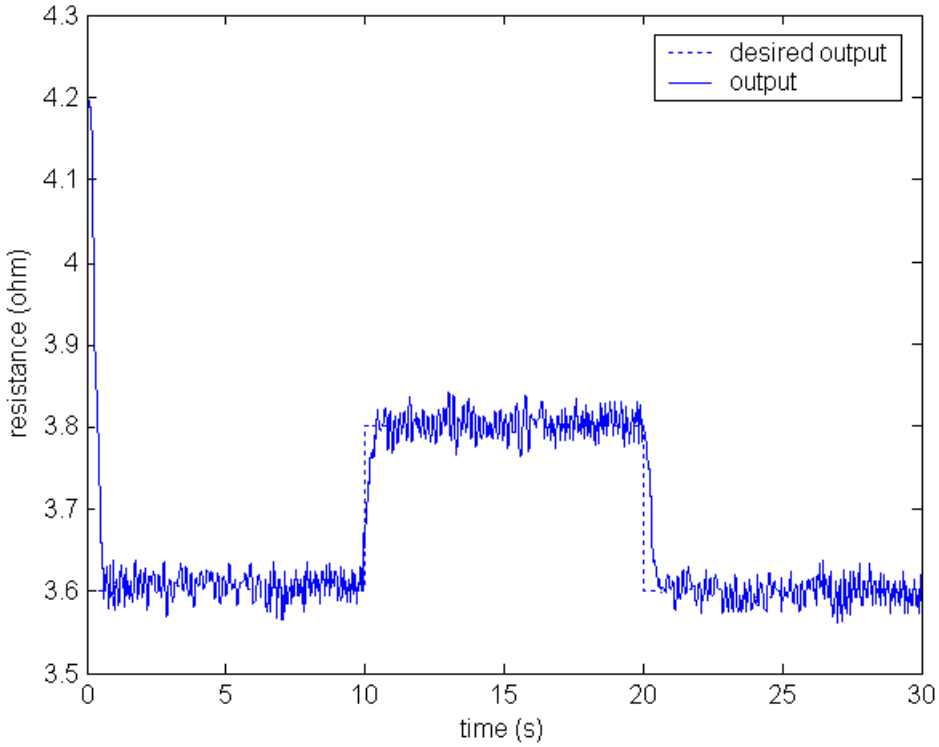


Figure 6-10 Step responses of the control system

Fig. 6.10 has demonstrated that the control system has well maintained the output resistance of the SMA to the desired output. Although the system is compensated with PID controller, the noise still exists. The future work should consider to improve in the control system with the consideration of the observation noise.

6.8 Conclusion

In conclusion the hardware for the Nitinol Actuated Stewart Platform has been successfully designed and constructed. This chapter has given an overview of the circuit board layout and the details have been explained clearly. A brief control strategy was also presented along this chapter. The alternative control system does not require a special mechanism, such as measurement of inner variations such as temperature or resistance. Hence its application to micro-SMA is expected to result in improved performance. However it needs to improve in the control system with consideration to observation noise, and it has considered that the problem which it should solve is expectable in the control system which compensates the hysteretic property of a certain thing.

Chapter 7 First Prototype

7.1 Required modifications and improvements

A number of small modifications and improvements were carried out on the Stewart Platform to improve its operation. The major reason for the modifications to the final design was due to the manufacturing tolerances were not close enough. Some areas still need to be addressed to get the Nitinol Actuated Stewart Platform performing as well as intended.

- The snap joints on each of the prismatic triangular box did not provide sufficient strength from holding the other half of the box, causing each module of the snake robot to fall apart when placing the snake robot on an inclined surface. To maintain the rigidity of the design, each corner of the prismatic triangular box was drilled with three holes and was tapped with M2 threads (See Appendix A for more details of the holes). These holes were for the M2 cap screws to hold each module of the Stewart Platform tightly.
- The roundness of the edge of each ball joint cone was not consistent, causing problems when attempting to fit it into the ball joint inner housing located along the side of the prismatic boxes. This problem was overcome by sanding the ball joint cones edges with sand paper to the right roundness for the corresponding ball joint inner housing.
- Initially the guiding slots for supporting the female connectors were grown with shoulder support feature to fit the female connectors into place. However this feature was way too small to be manufactured by Rapid Prototyping. Each slot was left with excessive material built up rather than the feature shoulder support. The excess materials were removed by using M1.6 drill.
- The holes where located at both ends of each leaf spring did not have the same consistency of depth. These holes were meant to hold the ball connection rod into place (See Fig 3.4). This problem could be tolerated by having a longer connection rod for the one which has deeper depth and vice versa.
- An additional insulation tube was used to cover the connection between the electrical wire and the crimp. This should strengthen the connection point.

After carrying out the modifications listed above and connecting the assembled circuit boards into the active stages, the Nitinol Actuated Stewart Platform was ready for use. The tests carried out are described in the following sections.

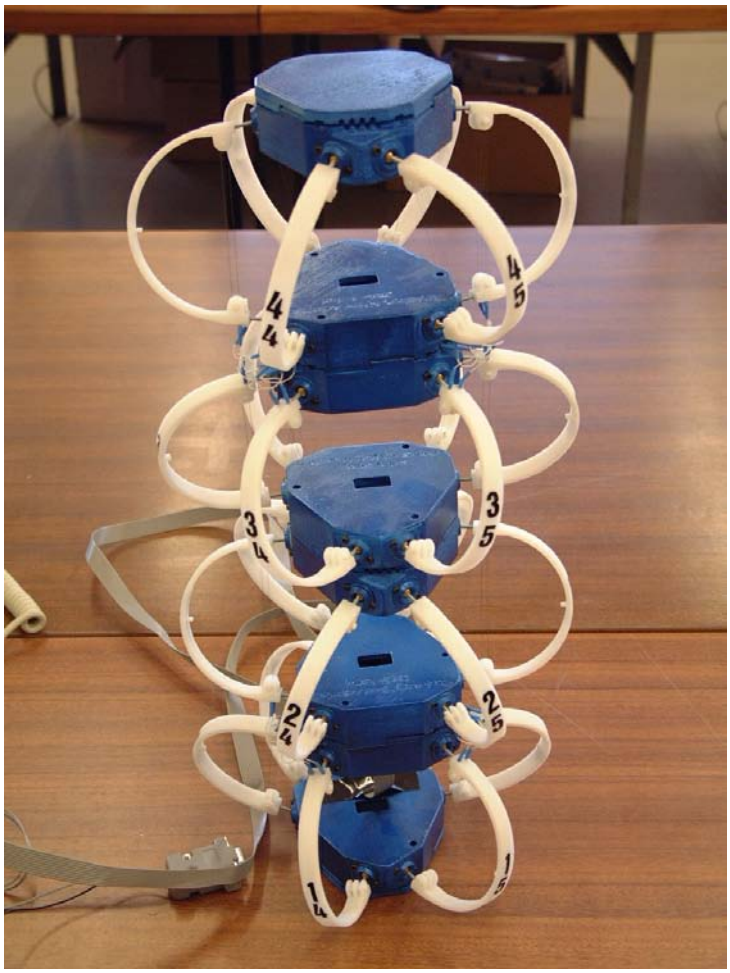


Figure 7-1 A four stages nitinol actuated Stewart Platform

7.2 Measured Performance of Nitinol actuated Snake Robot

A set of criteria were developed to try and quantify the performance of the Nitinol Actuated Steward Platform so we could compare the results obtained against our results presented in Chapter 4.

7.2.1 Performance Criteria

The important criteria for the Nitinol Actuated Stewart Platform were deemed to be:

- Maximising contraction force and bending force output
- Maximum speed of operation.
- To be able to perform complex motion. Meaning that it can alter it shape of its own accord over its whole length.

Force output is quite easily measured using a spring balance, and speed of operation can be timed reasonably well with a stopwatch. Mobility of the Nitinol Actuated Stewart Platform is a much

easier parameter to measure. Again size and weight are easy to measure, but a test of power consumption, especially for a SMA actuated robotic device that is not performing a set of predefined tasks like most of the conventional robot often would be, is a little more involved. When all the actuators are off the power consumption is 0, and if the ambient temperature was very low (requiring 100% PWM), then the power is 4 x 6 x (18V x 0.015A x PWM) = 7.3W. Aesthetic appeal is very much up to the individual, but here we will take this as meaning how good the surface finishes compare with the traditional robotic device using metal or aluminium as their main body frame.

7.2.2 Test Procedure

Measurement of force output

The base of the structure was clamped on to a table and hooked to a spring balance in different orientations, as shown in Figs 7.1, 7.2 and 7.3. The first test was to testify how much force could a single stage of the Nitinol Actuated Stewart Platform generate when all the actuators within the top stage were contracted. Followed by a second test, to testify the entire structure's pulling force in the vertical axis. This required the entire structure's actuators contracted all at once by applying maximum PWM to all of the actuators. The set up for both testings would require a spring balance hooked to the top deck, whereas the spring scale was to be hung vertically with a stand. During these testing a stopwatch was used to time the speed of each operations. The layout is shown as follows. The change in lengths were labelled as L_1 and $L_{2,}$ These lengths were to be used for further research enquires.

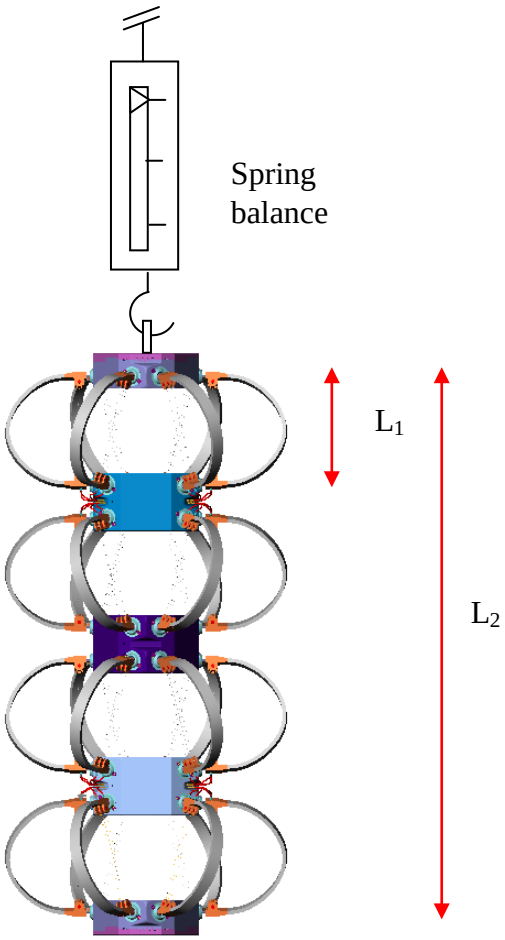


Figure 7-2 Set up for force testing in vertical axis.

After measuring the vertical contraction force, then bending force was tested by two different settings, as shown in Figs 7.2 and 7.3. The 3rd set up was to rotate the spring balance about the hook to where it was lying parallel with the top cover. The third set up was to have the spring balance over an actuator rather than between them. To maintain the spring balance in the air a clamp was used to hold the spring balance body. Notice that the two actuators shown producing the force have to be adjacent to each other. Then apply maximum PWM to the two actuators of each stage that are on the opposite side to the spring balance. Although both bending forces F_1 and F_2 were acting in different directions, the resultant bending force acts directly in the opposite direction to the spring scale. Then directly measure the resultant force off the spring scale and the distance that it pulled. See figure below.

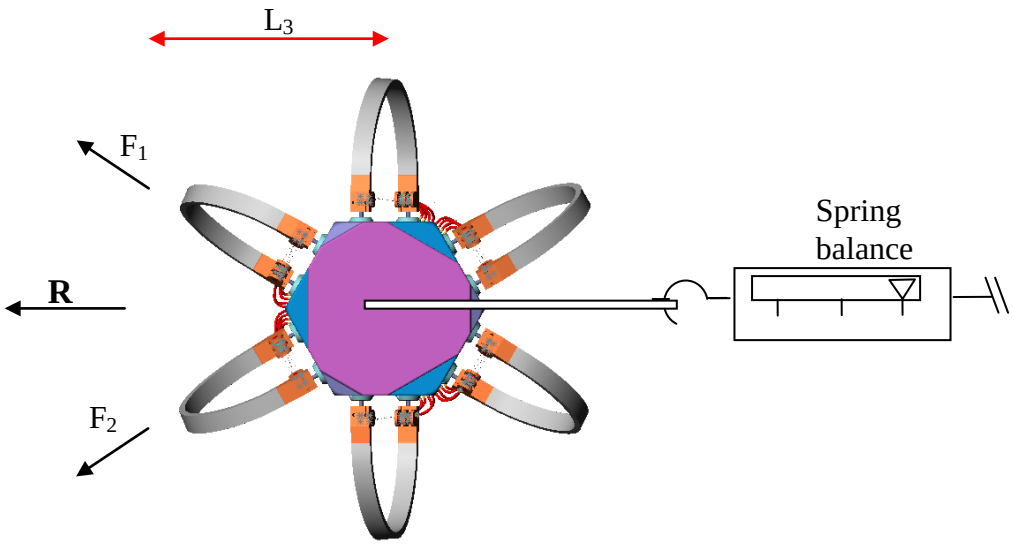


Figure 7-3 Set up for bending force testing along a horizontal axis in orientation one.

The last test was almost exactly the same as the previous set up except the spring balance is pulling against only one row of the actuators. Then apply a maximum PWM to every stage of the actuators, which were located in the opposite side of the spring scale. The purpose of running these two methods for testing the bending force was to compare the difference between the force that shared two columns of actuators and the force that generated by a single column of actuators. Each of these tests was repeated five times to get an average force output in each case.

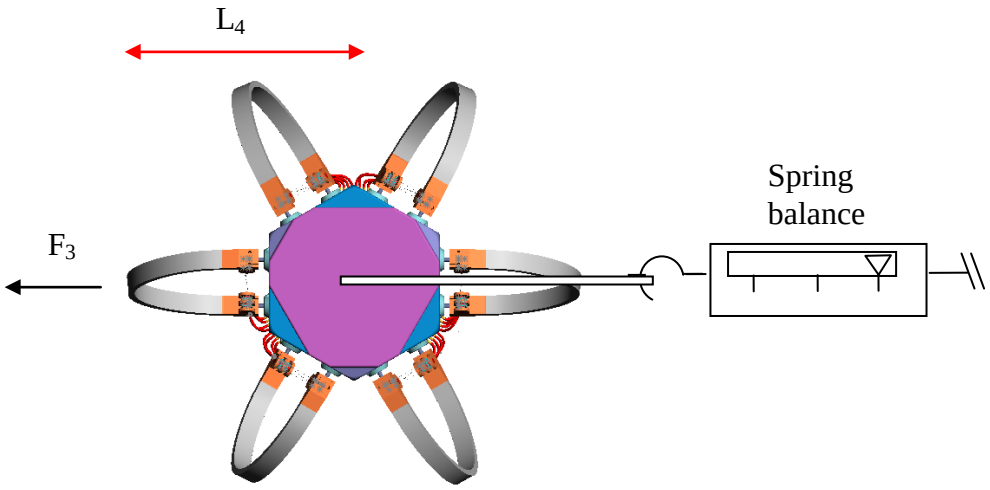


Figure 7-4 Set up for bending force testing along horizontal axis in orientation two.

Measurements of the speed of operation

A stopwatch was used to measure the time that it took for all of the actuators fully contracted with full current applied and when it stretched to its' initial form.

7.3 Results

Force Output

The average force and torque obtained for the four tests were 13.8N, 61.2 N, 2.4 Nm and 1.53 Nm respectively. Whilst the change in length L_1 to L_4 were 4.4 mm, 17 mm, 18.5 mm and 18.2 mm respectively.

Speed

Theoretical according to the technical data provided by Dynalloy, a 0.003" nitinol wire would require 1 second to fully contract and about 0.5 second to stretch to initial length. Closure times of approximately 1 second for contraction and 1.5 second for cooling were observed in the laboratory.

Weight

The finished Nitinol Actuated Stewart Platform weighs 0.7 kg, not including power supplies.

Size

The whole robot consisting of the four Stewart Platform stages is about 443 mm tall. Its maximum width across the leaf springs is 195 mm.

7.4 Discussion

Force Output

The force that a typical colonoscope could generate is about 1.5N with a weight of 32g but the Nitinol Actuated Stewart Platform produces 61.2 N total contraction force and 2.4 Nm bending force. Also the colonoscope bending force can only be generated at the free end. Moreover since the structure itself consists of an elastic element as the main body support, the elastic element is a hollow incompressible tube free to flex. Thus the colonoscope can only move within the coronal plane. However these limitations do not applied to the Nitinol Actuated Stewart Platform. It is not limited to the end section from generating force, but also each individual module along its body. In addition it is free to contract along the vertical axis. Moreover it weighs 704.3 g so its strength to weigh ratio (87.4

N/kg) is more than two times that of the colonoscope. Also the prototype multistage Stewart Platform produced during this research can be scaled in size to suit the application. Thus the strength to weigh ratio could increase.

Speed

The experiments results, average around 1 second for one module to contract fully and 1 second for it cool to room temperature. The time required for the whole 4 stage Stewart Platform to contract is about 1 second and 1.5 seconds for it to stretch back to its original form. The operation time would be less if the Stewart Platform is in a cooler environment.

Weight

A single stage of the Stewart Platform has an average weight of 165.4 g, where the entire 4 stages weighs 704.3g (include both base and top cover). This is heavy compared to a colonoscope that weighs 32g. Depending on application the Stewart Platform produced during this research can be scaled in size to suit. Thus the weight would depend on the task. For example, an appropriately sized multistage unit could inch along the inside of pipe work in either direction.

Size

The size of the finished nitinol actuated Stewart Platform is quite bulky, as can be seen in Figs 7.1 and 7.2. It is big in size because of the leaf springs and the excessive spacing in each prismatic triangular box. The future design should address to minimise the thickness of the prismatic triangular box. Also it would require a redesign of the leaf spring which could be constructed as a zig zag. A lot of development work is needed if the Nitinol Actuated Stewart Platform is going to match the size of a colonoscope. Please refer to Appendix A for detail dimensions.

Aesthetic appearance

Apart from the surface finish, the robot is interesting to look at. As expected the "stairstepping" effect produced by the Rapid Prototyping (See Chapter 3) is visible. The surface roughness can be overcame by spraying a layer of automotive primer evenly on the surface of the parts and then smoothing it with sand paper. Eventually the automotive primer fills up the spacing that caused by the “stairstepping” effect. Then the parts can be painted with colour paint to improve the appearance.

Dexterity

With a total of 24 independent dof for the four stages Nitinol Actuated Stewart Platform, there are 224 possible positions, many more than most of the nitinol actuated mechanisms reviewed. However the workspace was not as large. This is because most of the other nitinol actuated mechanisms usually consisted of multiple sections to perform bending at the same time. Consequently the bending angle of the tip would be amplified by each of the section along the structure. Therefore the higher number of sections of the nitinol actuated mechanism consist the larger bending angle of the tip and the wider work space through amplification (although at reducing force). However with the unique design of the Stewart Platform, additional sections could be added by stacking it on top of the current structure. Alternatively replacement of nitinol wires to nitinol springs would obtain a greater range of motion and higher force output. But this in turn would pose a problem of slow response time from cooling the nitinol spring to ambient temperature.

7.5 Conclusion

A prototype nitinol Actuated Stewart Platform has been built. A number of minor modifications were made to the Stewart Platform after it was first assembled, and a simple tests were carried out to evaluate the performance. Test results show that it produces a maximum tip force of 61.2 N and will travel through its full range of movement in approximately 1 s and take 1.5 s for it to retain its initial position. A number of areas need to be improved in order to approach the performance of colonoscopes or other nitinol actuated mechanisms.

However a simple structure has been completed for evaluation purpose and is capable of 10 million position states. It should be noted that some of these states may be very similar.

Chapter 8 Summary and Conclusion

8.1 Summary of Work

The objective of this thesis was to design, built and test a multi stage Stewart Platform actuated by shape memory alloy. To fulfil these objectives, research was carried out to investigate the properties of the shape memory alloy to better understand its behaviour. Nitinol was chosen as the actuator, because shape memory alloys based on nickel and titanium have, to date, provided the best combination of material properties for most commercial applications. To ensure the chosen actuator was suitable for the task, a number of different test methods were carried out. Although some methods were unsuitable for testing the nitinol wire, experience was gained through these different testing methods. Eventually a simple and effective experimental set up was developed for experimenting the nitinol wire, and a strain feedback control developed

The specimen was a Ti-Ni SMA wire, 0.762 mm in diameter and 113 mm in length, produced by Dynalloy, Inc. The experimental set up was no more than just a conventional Wheatstone bridge supported by a quad transistor pack. This quad transistor pack provides good temperature compensation for extracting useful data. By driving the wire with different percentages of PWM, the wire would be heated accordingly. Along with the Wheatstone bridge, an encoder was also used to measure the corresponding strain. Results were then collected and analysed. The results showed that the expected hysteresis associated with martensitic transformation; This hysteresis was proven to have a linear relationship with resistance, length and volume. This characteristic has useful applications for inherent position and force detection.

Once the properties of the SMA wire were understood, the SMA limitations were the next problem that needed to be solved. There are two main technical problems. One is the low energy efficiency in transforming electrical power into strain in the SMA actuator. In general, the efficiency of a conventional robot system is about 40-50% while that of the SMA robot is only 5-6%. One way to improve the energy efficiency is to a larger diameter SMA wire instead of use multiple smaller diameter nitinol wires placed in parallel to increase the lifting capabilities of an SMA actuator. This reduces heat loss (volume/Area is reduced) but loses the benefit from the higher cooling speeds of multiple smaller wires while obtaining the high force from the actuator. A second problem is the fatigue and shape memory degradation following the continued use of an SMA actuator. Improvements in mechanical design may be necessary, but this may not be a major technical problem considering that SMA Ni-Ti materials have relatively long fatigue lives.

The transient conduction of the SMA wire was also studied. The temperature gradient within the SMA has a side effect – internal strain. Since the SMA wire diameter is so small, the temperature gradient is almost negligible. Besides the Biot number was worked out to be less than 0.1, so determination of the temperature gradient was unnecessary.

After characterising of the SMA wire, the Stewart Platform was designed. Due to the complexity of the design and the many common parts, it was decided that the Stewart Platforms would be manufactured by rapid prototyping. Rapid prototyping is a fast and cost effective way to produce a prototype product. Although it is suppose to be used by designers and engineers to produce examples of their concepts for early visualisation, verification, iteration and optimisation, the complexity of the final design make it too expensive to be built through conventional machining methods. Hence Rapid Prototyping was used to produce the required parts.

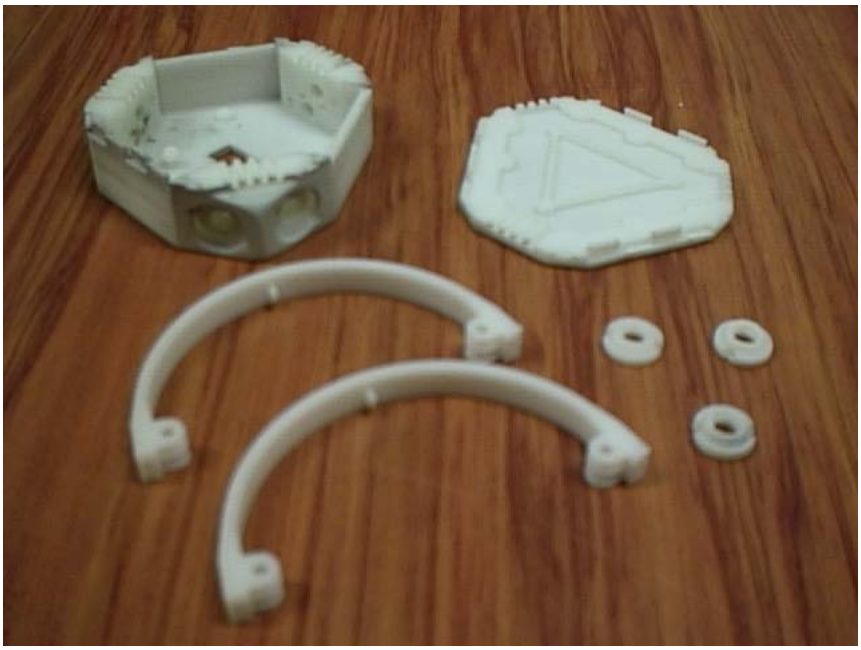


Figure 8-1 Stewart Platform parts.

The final design of the multistage nitinol based Stewart Platform mechanism consisting of a number of single stage Stewart Platforms is shown in Fig 1.3. Each end piece of the Stewart platform was connected by 6 parallel bows that are preloaded to give a spacing of 80 mm at ambient temperature. The Stewart Platform was designed specifically as a hexagonal box shape. The internal space of the triangular box was used for holding and protecting the circuit board. The length of the triangular box is about 100 mm, and the height when the box closed is about 40 mm. Slots are available in the edges of the box for the connection of the nitinol wire to the circuit board.

Each stage of 6 nitinol actuators has six degrees of freedom and a total of 64 combinations of movement. Hence the multi-stage mechanism should be capable of performing quite complex movements. With the ball joint provided at the end of the bow, rotation of the leaf spring is possible, but this does not affect the attitude of the platform, and the spring can be elastically constrained if desired.

With the finished construction of the prototype Nitinol Actuated Stewart Platform, a number of tests were carried out to check the performance of the Stewart Platform. The collected results were then compared with the nitinol actuated colonoscope. The comparison was based on a set of predefined criteria. It was discovered that the workspace was not as large as desired since the SMA material had to be "trained" and the recoverable strain was reduced 40%. To over come this difficulty, it is suggested more decks be built to amplify the tip workspace or else to replace nitinol wires with slower nitinol springs. Given the unique design of the Stewart Platform, this suggestion can be easily accomplished. Overall the performance was satisfactory and the robot characteristics could be checked.

8.2 Conclusion

The Nitinol Actuated Stewart Platform was successfully developed. With 2 microcontrollers, and 4 Stewart Platform stages, 2^{24} poses are possible, and distributed uniformly over 6 degrees of freedom, there are 16 possibilities for each dof. With 8 sections there are 256 possibilities. The robot is able to perform complex motion as expected, but the SMA strain is not high enough to produce a large working envelope. SMA's can only achieve a workable strain of 5 percent and this is reduced by training. There may be alternative alloys that have even higher percentage of strain that are under development, and certainly there are some high voltage polymers that show strain rates around 50%. Future work may entail a more rigorous development of another SMA/polymer actuator.

The forward kinematics were developed for a single stage of the Stewart Platform. Using small perturbations. Future work may entail developing a general model for the inverse kinematics. With this groundwork for forward kinematics, for small strains is a useful contribution. A rigorous singular position analysis of the mechanisms has not been carried out in this thesis. But has been done else where [6].

Since each end of the leaf spring is attached to a ball joint, the leaf springs are free to yaw about the 2 ball joints. Once it has been rotated, it is unlikely that it can retain to its initial position. Thus future modification should address this, for aesthetic reasons.

A simple modelling method was introduced in this thesis to estimate the leaf spring's stiffness. The prototyping methods were unique in that they add and bond materials in layers to form objects. Thus given the same material, a property such as elastic modulus E is no longer the same as models formed by molding. For this reason a matlab program was been written to model the composite. Time constraints prevented the program from being fully utilised, but it is available for future work (see Appendix C). In order to give a better performance, it was suggested that the leaf spring should be remodelled in future, perhaps as a zig zag unit that can still take advantage of the rapid prototype layering.

No rigorous optimisation of the geometry has been carried out in this thesis. Future work could entail finding optimum ratio between the circuit board size, the platform size and the leaf spring size. This work would probably need to be done in conjunction with the stiffness/deflection modelling for the leaf springs so that the best ratio can be found to minimise the compliance of the mechanism.

Due to the time constraints, a simple control program of the Nitinol Actuated Stewart Platform was written. However the hardware has successfully built, and the resistance measurement feedback control was tested. A control strategy has also presented in the thesis. However it needs to improve in the control system with consideration to observation noise. Moreover a more optimal control programme is needed to be written for further development.

In conclusion, the Nitinol Actuated Stewart Platform was designed, built and successfully tested. Rapid prototyping was shown to be a viable means for simple and quick fabrication of prototypes for articulated structures such as robotic systems. The successful fabrication of the Stewart Platform gives further confidence in this Rapid Prototyping framework. A paper has been published on the work and it has led to several interesting avenues for further research

Appendix A Engineering drawings of the robot components

This appendix contains the detailed engineering drawings of the components.

Drawing No. A.1: Canterbury Nitinol Snake Robot

Drawing No. A.2: Base Termination

Drawing No. A.3: Circuit Board

Drawing No. A.4: Platform Section

Drawing No. A.5: Ball Joint Cone

Drawing No. A.6: Leaf Spring

Drawing No. A.7: Assembled Leaf Spring

Drawing No. A.8: Brass Ball Bearing

Drawing No. A.9: Threaded Bar

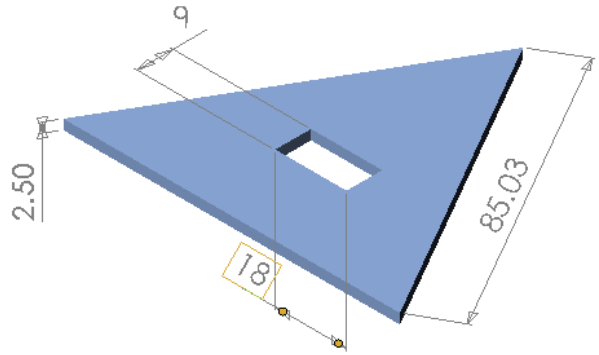
Drawing No. A.10: PTFE

Appendix B Weight Estimation for the Manipulators

B.1 Estimation of the Platform weight

This section is demonstrating a rough idea of how does the weight of each platform is calculated. To ease the calculation, some fine details of the platform are ignored.

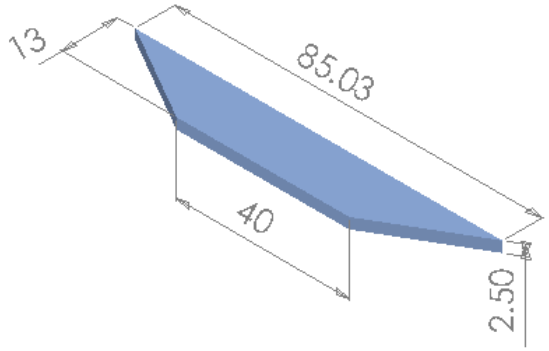
B.1.1 Platform base:



$$\text{Area} = \frac{1}{4}\sqrt{3}(0.08503)^2 - (0.009x0.018)$$

$$= 2.97 \times 10^{-3}\ \text{m}^2$$

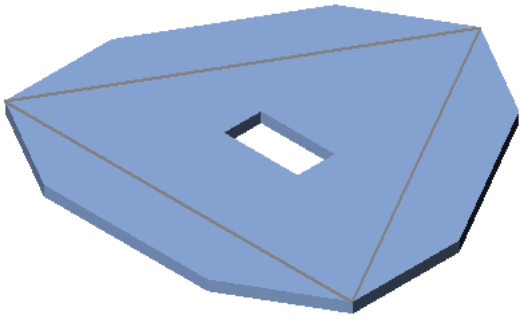
Figure B. 1 An equilateral triangular base with the ribbon feature.



$$\text{Area} = \frac{(0.04 + 0.085033)x\ 0.013}{2}$$

$$= 8.127x\ 10^{-4}\ \text{m}^2$$

Figure B. 2 Parallelogram side.



$$\text{Area} = 3 \times 8.127x\ 10^{-4} + 2.97 \times 10^{-3}$$

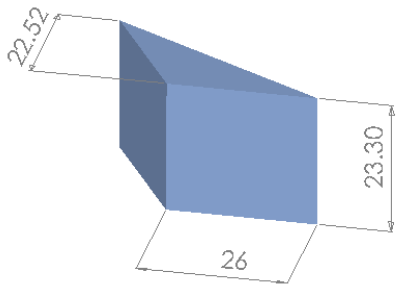
$$= 5.41x\ 10^{-3}\ \text{m}^2$$

$$\text{Volume} = 5.41 \times 10^{-3} \times 0.0025$$

$$= 1.352 \times 10^{-5}\ \text{m}^3$$

Figure B. 3 An equilateral triangular base joined with three parallelogram sides.

B.1.2 Ball joint housing

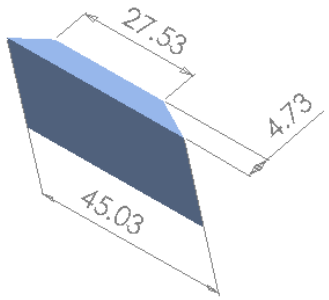


Area of the triangular surface = 0.5 x 0.02252 x 0.026

= 2.928 x10^{-4} m^2

Figure B. 4 A prismatic triangular shape.

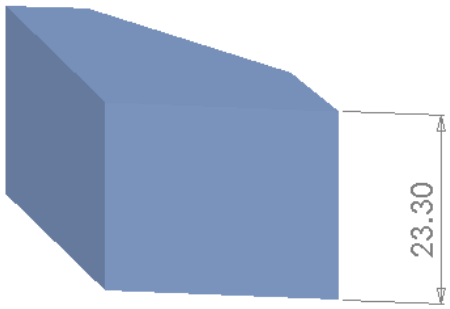
Part 2



Area of the parallelogram surface =

$$\frac{(0.0275+0.04503)x\ 0.00473}{2} = 1.717 \times 10^{-4}\ m^2$$

Figure B. 5 A prismatic parallelogram shape.



Total area of the ball joint housing

= 1.717 x 10^{-4} +2.928 x10^{-4} = 4.645 x 10^{-4} m^2

Volume = 4.645 x 10^{-3} x 0.0233 = 1.082 x 10^{-5} m^3

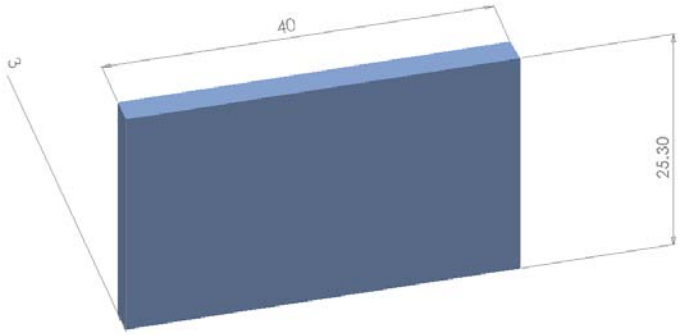
Figure B. 6 Final form of the ball joint housing.

Take out the area of the ball joint.

Volume of the ball joint = $\frac{4}{3}\pi(0.00395)^3$ =2.581 x 10^{-7} m^3

Assuming half of the volume of the ball joint goes into the housing. Final volume of the ball joint housing = 1.082 x 10^{-5} – 2.581 x 10^{-7} = 1.056 x 10^{-5} m^3

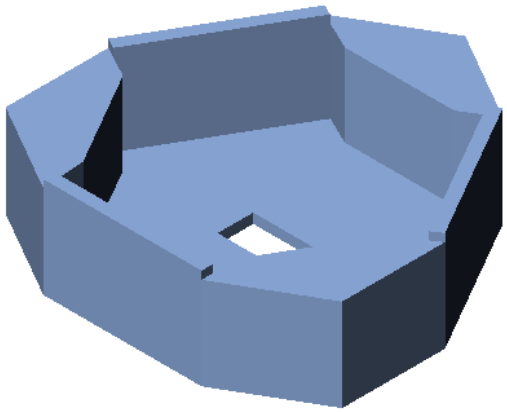
B.1.3 Side wall



Volume = 0.04 x 0.003 x 0.0253

$= 3.036 \times 10^{-6}\ m^3$

Figure B. 7 A prismatic side wall.



Final volume =

$$3x\,(3.036 \times 10^{-6} + 1.056 \times 10^{-5}) + 1.352 \times 10^{-5}$$

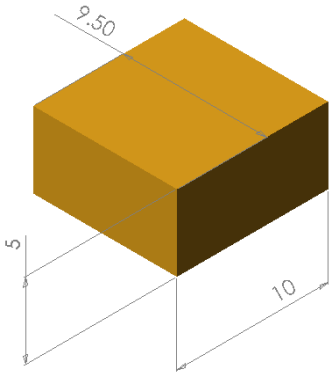
$= 5.432 \times 10^{-5}\ m^3$

Figure B. 8 Final form of a single deck of prismatic equilateral triangular boxes.

Estimate weight of a single deck platform = $5.432 \times 10^{-5} \times 903.703 = 0.049$ kg

B.2 Estimation of the leaf spring weight

B.2.1 The end section of the leaf spring:

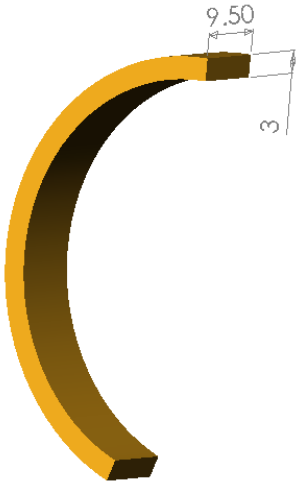


Volume = 0.0095 x 0.005 x 0.01

= 4.75×10^{-7} m^3

Figure B. 9 A prismatic rectangular shape.

B.2.2 Body of the leaf spring:



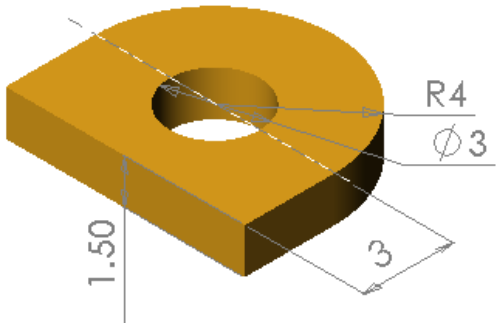
The outer radius is 0.0465m.

Volume = $(\pi \times 0.0465^2)-(\pi \times 0.0435^2) \times 0.0095 x \frac{160}{360}$

= 3.58×10^{-6} m^3

Figure B. 10 The body of the leaf spring.

B.2.3 Teflon rod slot:



Volume = $((0.5 \times \pi \times 0.004^2) + (0.003 \times 0.008) - (\pi \times 0.0015^2)) \times 0.0015$

$= 6.3 \times 10^{-8}\ m^3$

Figure B. 11 The Teflon rod slot.

Final volume of the leaf spring = $(2 \times 4.75 \times 10^{-7}) + (6 \times 6.3 \times 10^{-8}) + 3.58 \times 10^{-6}$

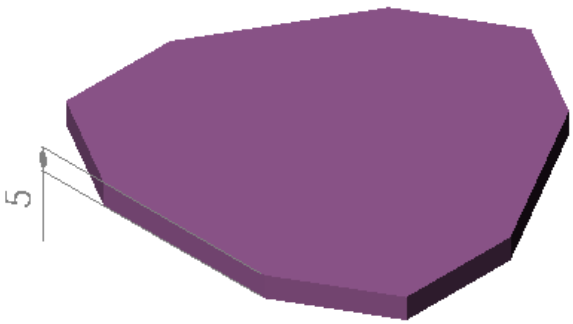
$= 4.908 \times 10^{-6}\ m^3$

Estimate weight of the leaf spring = $903.73 \times 4.908 \times 10^{-6}$

$= 4.435 \times 10^{-3}$ kg

B.3 Estimation of the base cover

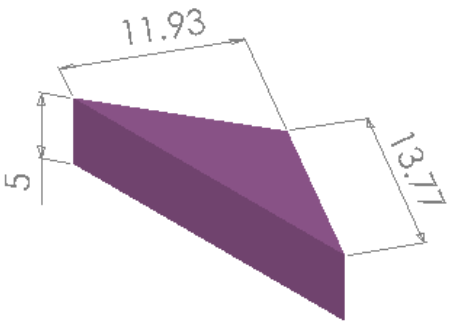
Assume the base cover has got the same shape as the base of the platform section, except the ribbon hole feature.



Area = $5.41 \times 10^{-3} + (0.009 x 0.018)$

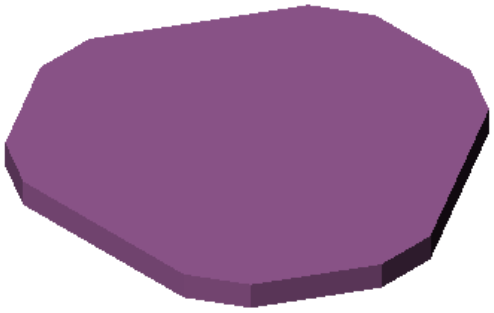
$= 5.572 \times 10^{-3}\ m^3$

Figure B. 12 The initial shape of the base cover.



Area of the triangular surface
= 0.5 x 0.01193 x 0.01377

Figure B. 13 The truncated corner.



The final area
= $5.572 \times 10^{-3} - 3 \times 8.214 \times 10^{-5}$
= 5.818×10^{-3} m^3

Final weight
= $903.703 \times 0.005 \times 5.818 \times 10^{-3}$
= 0.0263 kg

Figure B. 14 The final form of the base cover.

B. 4 Miscellaneous weight

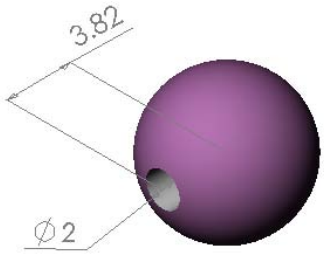


Figure B. 15 Brass ball bearing.

The density for brass is 8800 kg/m^3. The volume for each brass ball bearing is 2.581×10^{-7} m^3. Thus the weight for each ball bearing is 2.72×10^{-3} kg. Subtract the volume of the hole with the dimension of ϕ2mm x 3.82 mm. The final weight for each brass ball bearing is 2.166×10^{-3} kg.

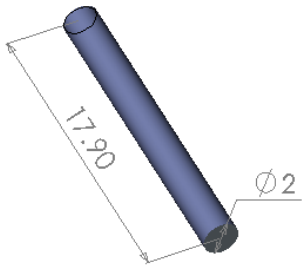


Figure B. 16 A ϕ2 mm x 17.9 mm threaded rod.

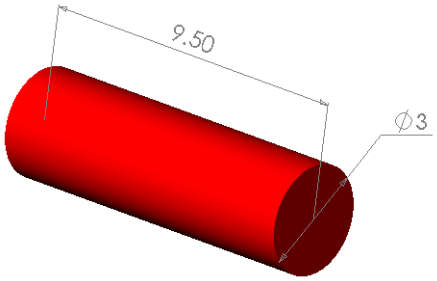


Figure B. 17 A ϕ3 mm x 9.5 mm Teflon rod.

The weight for the ϕ2 mm x 17.9 mm threaded rod is 4.386 x 10^{-4} kg, where the weight of the ϕ3 mm x 9.5 mm Teflon rod is 9.267 x 10^{-5} kg.

Appendix C Matlab Code for Optimising Leaf Spring

This appendix contains the program suggested to be use for optimising the leaf spring.

C1.cltc_ui.m

```
function [] = cltc_ui()

% CLTC_UI user interface for composite calculation program
%
% written by: Angelo Garcia

% ---------------------------------------------------------------------
% Program variables

% Change current working directory to the directory where all CLTC
% files are located. This is useful when dealing with .mat files in
% which the material properties are stored.
% .mat files have to be in the same directory as all program files.


uifilename = 'cltc_ui.m';
uifilenamelength = length(uifilename);
cltcpath = which('cltc_ui.m');
cltcpathlength = length(cltcpath);
cltcpath(cltcpathlength-uifilenamelength:cltcpathlength) = [];
cd(cltcpath);

% load .mat files with material properties
load('cltc_data.mat');

% ---------------------------------------------------------------------
% Main Menue, select what to do

while 1==1
   % clc
   disp(' ')
   disp('*********************************************************************')
   disp('*      CLTC - Classical Laminate Theorie Calculations       *')
   disp('*         written by: Michael Schlotter, May 2002          *')
   % disp('*                                                          *')
   % disp('*             written by: Angelo Garcia                    *')
   % disp('*                    May 2002                      *')
   disp('*********************************************************************')
   disp(' ')
   disp('Select from the list by typing in the number at the left:')
   disp(' ')
   disp('* CALCULATIONS:')
   disp('(01): Calculate the ELASTIC CONSTANTS OF A COMPOSITE MATERIAL')
   disp('     Optional: save it in the list of predefined composites')
   disp('     -> Matrix and fibre properties should be predefined')
   disp(' ')
   disp('(02): Calculate the TRANSFORMED ELASTIC CONSTANTS OF A LAMINATE')
   disp('     Optional: save it in the list of predefined laminates')
   disp('     -> All used composite materials should be predefined')
   disp(' ')
   disp('(03): Calculate the STRAINS IN A LAMINATE with given Force and')
   disp('     Moment Resultants')
   disp('     -> The laminate must be predefined')
   disp(' ')
   disp('* LISTS OF PREDEFINED MATERIALS:')
   disp('(11): See list of predefined MATRIX materials')
   disp('(12): Enter new matrix material manually')
   disp('(13): Delete a matrix material from the list')
   disp(' ')
   disp('(21): See list of predefined FIBRE materials')
   disp('(22): Enter new fibre material manually')
   disp('(23): Delete a fibre material from the list')
   disp(' ')
   disp('(31): See list of predefined COMPOSITE materials')
   disp('(32): Enter new composite material manually')
```

```
disp('(33): Delete a composite material from the list')
disp(' ')
disp('(41): See list of predefined LAMINATES')
disp('(42): Enter new laminate manually')
disp('(43): Delete a laminate from the list')
disp(' ')
disp('(00): Options: save lists, exit program...')
disp(' ')
usersel = input('Type in a 2 digit number: ');

% clc
disp (' ')

switch usersel
case {01}
   % input volume fibre fraction
   fib_frac = abs(mod(input('Enter Volume Fibre Fraction: '),1));
   fprintf('-> Volume Fibre Fraction: %2.4f \n',fib_frac)

   % call input- subfunction for matrix and fibre
   matinput = input_matfib(data_matrix,'matrix');
   fibinput = input_matfib(data_fibre,'fibre');

   % call micromechanics calculation function
   [e1,e2,g12,nu12,nu21] = cltc_micro(matinput, fibinput, fib_frac);
   elprops = [e1, e2, g12, nu12];
   fprintf('\nThe new composite material has the following properties: \n')
   fprintf(['* Elastic Constants: \n' ...
         '  E1: %4.2e Pa \n  E2: %4.2e Pa \n  G12: %4.2e Pa\n'...
         '  nu12: %6.4f\n  nu21: %6.4f \n\n'],e1,e2,g12,nu12,nu21)
   fprintf('* Fibre Volume Fraction: \n  Vf: %4.2f \n\n',fib_frac)

   % call subfunction to save new composite
   data_comp = new_calcmat(data_comp,elprops,fib_frac,'composite');

case {02}
   % call input- subfunction for plies
   matprop = input_ply(data_comp);

   % call input- subfunction for angles and thichnesses
   lam_angle = input_thiang(size(matprop,1),'angles','degrees');
   lam_thick = input_thiang(size(matprop,1),'thicknesses','millimeters');
   lam_thick = lam_thick/1000;

   % call macromechanics calculation function
   [ex,ey,gxy,nuxy,nuyx,eta13,eta31,eta23,eta32,ABD,abcd] = ...
      cltc_macro(matprop, lam_angle, lam_thick);
   elprops = [ex, ey, gxy, nuxy];
   fprintf('\nThe new laminate has the following properties: \n')
   fprintf(['* Elastic Constants: \n' ...
         '  Ex: %4.2e Pa \n  Ey: %4.2e Pa \n  Gxy: %4.2e Pa \n' ...
         '  nuxy: %6.4f \n  nuyx: %6.4f\n' ...
         '  eta13: %6.4f \n  eta31: %6.4f\n' ...
         '  eta23: %6.4f \n  eta32: %6.4f\n\n'] ...
      ,ex,ey,gxy,nuxy,nuyx,eta13,eta31,eta23,eta32)
   format long g;
   fprintf('* Matrices: \n')
   fprintf('  ABD- Matrix: \n')
   disp(ABD)
   fprintf('  abcd- Matrix: \n')
   disp(abcd)
   format short g;

   % call subfunction to save new laminate
   data_lam = new_calcmat(data_lam,elprops,ABD,'laminate');

case {03}
   % call subfunction to enter laminate and loads
   [lam_numb, load] = input_load(data_lam);

   % call strain calculation function
   if ~isempty(load)
      [e0x, e0y, g0xy, k0x, k0y, k0xy] = ...
         cltc_strain(data_lam(lam_numb).ABD, load);
      fprintf('\nStrains and curvatures at the reference plane: \n')
```

```
            fprintf(['* Stains: \n' ...
                  '  e0x: %6.4e \n  e0y: %6.4e \n  g0xy: %6.4e \n\n' ...
                  '* Curvatures: \n'...
                  '  k0x: %6.4e \n  k0y: %6.4e \n  k0xy: %6.4e \n\n'] ...
                 ,e0x, e0y, g0xy, k0x, k0y, k0xy)
         end
         msg_wait;

      case {11}
         fprintf('The predefined matrix materials are: \n')
         cltc_sdisp(data_matrix,1)
         msg_wait;

      case {12}
         data_matrix = new_mat(data_matrix,'matrix');

      case {13}
         data_matrix = delete_mat(data_matrix,'matrix');

      case {21}
         fprintf('The predefined fibre materials are: \n')
         cltc_sdisp(data_fibre,2)
         msg_wait;

      case {22}
         data_fibre = new_mat(data_fibre,'fibre');

      case {23}
         data_fibre = delete_mat(data_fibre,'fibre');

      case {31}
         fprintf('The predefined composite materials are: \n')
         cltc_sdisp(data_comp,3)
         msg_wait;

      case {32}
         data_comp = new_mat(data_comp,'composite');

      case {33}
         data_comp = delete_mat(data_comp,'composite');

      case {41}
         fprintf('The predefined laminates are: \n')
         cltc_sdisp(data_lam,4)
         msg_wait;

      case {42}
         data_lam = new_mat(data_lam,'laminate')

      case {43}
         data_lam = delete_mat(data_lam,'laminate');

      case {00}
         fprintf( 'Please select by pressing [key]: \n')
         fprintf(['- [s]ave all changes, made in material lists \n' ...
                  '- [e]xit and save all changes \n' ...
                  '- [q]uit without saving \n' ...
                  '- Any other key to return to the main menu \n\n'])
         usersel = input('Your choice: ','s');
         if (lower(usersel) == 's')
            save cltc_data.mat data*
         elseif (lower(usersel) == 'e')
            save cltc_data.mat data*
            break;
         elseif (lower(usersel) == 'q')
            break;
         end

      otherwise
         fprintf('Unknown choice! Please try again! \n\n')
         msg_wait;

      end
   end
```

```
% ********************************************************************
% subfuctions for calculation routines:
% user inputs and saving of calculated materials

function [userinput] = input_matfib(struct,material)
% INPUT_MATFIB input of elastic constants of fibre and matrix
while 1==1
   fprintf('\nChoose %s material from list: \n',material)
   cltc_sdisp(struct,0)
   fprintf('(0) Enter constant matrix manually \n\n')
   userchoice = input('Your choice: ');
   if userchoice == 0
      userinput = input('\nEnter row matrix [E, G12, nu12]: ');
      [m,n] = size(userinput);
      while (m~=1 | n~=3)
         fprintf('Wrong input! Please try again! \n\n')
         userinput = input('Enter row matrix [E, G12, nu12]: ');
         [m,n] = size(userinput);
      end
      break
   elseif ((userchoice>=1)&(userchoice<=size(struct,2)))
      userinput = struct(userchoice).elconst;
      break
   else
      fprintf('Unknown choice! Please try again! \n\n')
   end
end

% ---------------------------------------------------------------------

function [userinput] = input_thiang(numb_lay,value,unit)
% INPUT_THIANG input of thicknesses and angles for laminates
fprintf('\nDefine the %s of the layers in %s! \n',value,unit)
fprintf(['Column matrix: First value = layer 1, second value '...
      '= layer2, and so on. \nIf the values are equal in every '...
      'layer, enter just a single number! \n'])
userinput = input('Enter column matrix or number: ');
[m,n] = size(userinput);
while ~(m==numb_lay & n==1) & ~(m==1 & n==1)
   fprintf('Wrong input! Please try again! \n\n')
   userinput = input('Enter column matrix or number: ');
   [m,n] = size(userinput);
end
if m==1
   for k=1:(numb_lay-1)
      userinput = [userinput; userinput(1)];
   end
end

% ---------------------------------------------------------------------

function [userinput] = input_ply(struct)
% INPUT_PLY input of properties of the plies in the laminate
numb_lay = abs(fix(input('How many layers has the laminate? ')));
fprintf('-> Number of layers: %2.0f \n\n',numb_lay)

ip_hybrid = input('Interply hybrid laminate (y/n)? ','s');
while (~strcmpi(ip_hybrid,'n') & ~strcmpi(ip_hybrid,'y'))
   fprintf('Enter "y" or "n"! Please try again! \n\n')
   ip_hybrid = input('Interply hybrid laminate (y/n)? ','s');
end
if (lower(ip_hybrid) == 'n')
   fprintf('-> Interply hybrid laminate: NO \n')
elseif (lower(ip_hybrid) == 'y')
   fprintf('-> Interply hybrid laminate: YES \n')
end

k = 1;
while k <= numb_lay
   if (lower(ip_hybrid) == 'n')
      fprintf('\nChoose material for all layers from list: \n')
   elseif (lower(ip_hybrid) == 'y')
      fprintf('\nChoose material for layer %2.0f: \n' ,k)
   end
```

```
   cltc_sdisp(struct,0)
   fprintf('(0) Enter constant matrix manually: \n\n')
   userchoice = input('Your choice: ');

   if (lower(ip_hybrid) == 'n')
      if userchoice == 0
         fprintf(['\nThe elastic constants of all layers are equal. \n' ...
               'You must input a matrix with 4 columns and ' ...
               'one row. \n'])
         usermat = input('Enter row matrix [E1, E2, G12, nu12]: ');
         [m,n] = size(usermat);
         while (m~=1 | n~=4)
            fprintf('Wrong input! Please try again! \n\n')
            usermat = input('Enter row matrix [E1, E2, G12, nu12]: ');
            [m,n] = size(usermat);
         end
      elseif ((userchoice>=1)&(userchoice<=size(struct,2)))
         usermat = struct(userchoice).elconst;
      else
         fprintf('Unknown choice! Please try again! \n\n')
      end
      for loop=1:numb_lay
         userinput(loop,:) = usermat;
      end
      k = numb_lay+1;

   elseif (lower(ip_hybrid) == 'y')
      if userchoice == 0
         fprintf(['\nYou must input a matrix with 4 columns and ' ...
               'an arbitrary number of rows. \n' ...
               'The number of rows, added to the number of ' ...
               'already defined layers, must \n' ...
               'not exeed the  total number of layers! \n'])
         usermat = input('Enter matrix [E1, E2, G12, nu12]: ');
         [m,n] = size(usermat);
         while (k+m-1>numb_lay | n~=4)
            fprintf('Wrong input! Please try again! \n\n')
            usermat = input('Enter matrix [E1, E2, G12, nu12]: ');
            [m,n] = size(usermat);
         end
         userinput(k:k+m-1,:)=usermat;
         k = k+m;
      elseif ((userchoice>=1)&(userchoice<=size(struct,2)))
         usermat = struct(userchoice).elconst;
         userinput(k,:)=usermat;
         k = k+1;
      else
         fprintf('Unknown choice! Please try again! \n\n')
      end
   end
end

% ---------------------------------------------------------------------

function [userinput, load] = input_load(struct)
% INPUT_LOAD asks the user for laminate and applied loads
while 1==1
   fprintf('Choose laminate from list: \n')
   cltc_sdisp(struct,0)
   fprintf('(0) Cancel, the laminate is not listed yet! \n\n')
   userchoice = input('Your choice: ');
   if userchoice == 0
      fprintf(['\nPlease press (02) or (42) in the main menue to '...
         'define a \nnew laminate! \n\n'])
      userinput = [];
      break
   elseif ((userchoice>=1)&(userchoice<=size(struct,2)))
      userinput = userchoice;
      break
   else
      fprintf('Unknown choice! Please try again! \n\n')
   end
end
if ~isempty(userinput)
   fprintf(['\nEnter Force and Moment Resultants in N or Nm '...
```

```
        'respectively. \n'])
   load = input('Enter row matrix [Nx, Ny, Nxy, Mx, My, Mxy]: ');
   [m,n] = size(load);
   while (m~=1 | n~=6)
      fprintf('Wrong input! Please try again! \n\n')
      load = input('Enter row matrix [Nx, Ny, Nxy, Mx, My, Mxy]: ');
      [m,n] = size(load);
   end
   load = load';
else
   load = [];
end

% ------------------------------------------------------------------

function [new_struct] = new_calcmat(struct,elprops,auxprops,material)
% NEW_CALCMAT saves calculated materials in data- structures
fprintf(['Do you want to save the new %s material in the list \n' ...
      'of predefined composite materials? \n'],material)
newname=input('To save, enter a name; to cancel, just press [Enter]: ','s');
if (isstr(newname) & ~isempty(newname))
   newlength = size(struct,2)+1;
   struct(newlength).name = newname;
   struct(newlength).elconst = elprops;
   if strcmpi(material,'composite')
      struct(newlength).vf = auxprops;
      structdisp = 3;
   elseif strcmpi(material,'laminate')
      struct(newlength).ABD = auxprops;
      structdisp = 4;
   end
   fprintf('\nThe new %s material has be defined as following: \n',material)
   cltc_sdisp(struct(newlength),structdisp)
   msg_saverem;
else
   fprintf('\nThe new %s material has not been saved! \n\n',material)
end
new_struct = struct;
msg_wait;


% **********************************************************************
% subfuctions for data- structure modifications:
% entering new materials manually or deleting materials from lists

function [new_struct] = new_mat(struct,material)
% NEW_MAT adds a material to a data- structure
newlength = size(struct,2)+1;
fprintf('Enter name for new %s material',material)
newname = input(': ','s');
if (strcmpi(material,'matrix') | strcmpi(material,'fibre'))
   elconstmat = '[E, G12, nu12]';
   elconstsize = 3;
elseif (strcmpi(material,'composite'))
   elconstmat = '[E1, E2, G12, nu12]';
   elconstsize = 4;
elseif (strcmpi(material,'laminate'))
   elconstmat = '[Ex, Ey, Gxy, nuxy]';
   elconstsize = 4;
end
fprintf('Enter elastic constants %s',elconstmat)
newprop = input(': ');
[m,n] = size(newprop);
while (m~=1 | n~=elconstsize)
   fprintf('Wrong input! Please try again! \n\n')
   fprintf('Enter elastic constants %s',elconstmat)
   newprop = input(': ');
   [m,n] = size(newprop);
end
struct(newlength).name = newname;
struct(newlength).elconst = newprop;
if (strcmpi(material,'matrix') | strcmpi(material,'fibre'))
   structdisp = 1;
elseif (strcmpi(material,'composite'))
   newvf = abs(mod(input('Enter Volume Fibre Fraction: '),1));
```

```
   struct(newlength).vf = newvf;
   structdisp = 3;
elseif (strcmpi(material,'laminate'))
   newabd = input('Enter ABD- Matrix for new laminate: ');
   struct(newlength).ABD = newabd;
   structdisp = 4;
end
fprintf('\nThe new %s has be defined as following: \n',material)
cltc_sdisp(struct(newlength),structdisp)
msg_saverem;
new_struct = struct;
msg_wait;

% ------------------------------------------------------------------

function [new_struct] = delete_mat(struct,material)
% DELETE_MAT deletes a material from a data- structure
fprintf('The predefined %s materials are: \n',material)
cltc_sdisp(struct,0)
fprintf('(0) Cancel! I do not want to delete anything! \n\n')
while 1==1
   delitem = floor(input(['Enter the number of the material to ' ...
         'delete. You can also delete a couple of \n' ...
         'materials by entering a 1 by n matrix ' ...
         '[mat1, mat2,...]: ']));
   if delitem == 0
      fprintf('\nNo material has been deleted! \n\n')
      break
   elseif ((min(delitem)>=1)&(max(delitem)<=size(struct,2)))
      struct(delitem) = [];
      fprintf('\nThe %s material has been deleted \n\n',material)
      msg_saverem;
      break
   else
      fprintf('Unknown choice! Please try again! \n\n')
   end
end
new_struct = struct;
msg_wait;


% ********************************************************************
% subfuctions to display messages

function [] = msg_saverem()
% MSG_SAVEREM hint, how modified structures are saved on the harddisk
fprintf(['REMARK: The modified lists are not saved on your ' ...
      'harddisk until you \npress "00" in the Main Menue and '...
      'then [s] or [e] in the Options Menue! \n\n'])

% ------------------------------------------------------------------

function [] = msg_wait()
% MSG_WAIT just wait, until user presses a key. Then go to main menue.
fprintf('Press any key to get to the main menue \n\n')
pause
```

C2. cltc_macro.m

```
function [Ex,Ey,Gxy,nuxy,nuyx,eta13,eta31,eta23,eta32,ABD,abcd] = ...
        cltc_macro(prop,ang,thick)
% CLTC_MACRO determines laminate properties (macro mechanics)
%
% written by: Angelo Garcia
%
% Convention are as common in composite calculations.
% The rows in the input matrices correspond to the layers of the
% laminate as following:
%
% ============================ h(n)
%  layer n, ang(n), thick(n)
% ---------------------------- h(n-1)
%          .....                    /\
```

```
% ---------------------------- h(2)        /||\ z direction
%  layer 2, ang(2), thick(2)               ||
% ---------------------------- h(1)
%  layer 1, ang(1), thick(1)
% ============================= h(0)
%
%
% If the thickness matrix `thick' has one column, it represents the
% thickness of the layers, as shown above.
% If it has two colums, it is assumed, that it represents the
% thickness coordinates, measured from the reference plane in the
% middle of the laminate.
%
% thick = [ h(0)  , h(1);
%           h(1)  , h(2);
%             ...     ;
%           h(n-1), h(n) ]
%

% ---------------------------------------------------------------------
% some conversions and other stuff

% degrees to radians
ang = ang*(pi/180);

% number of layers = number of rows of input matrix
numb_lay = size(prop,1);

% enable variable input of thicknesses, as mentioned above
if size(thick,2) == 1
   height = sum(thick);
   hcoord(1,:) = [- height/2 , -height/2+thick(1)];
   for k = 2:numb_lay
      hcoord(k,:) = ...
        [-height/2+sum(thick(1:k-1)) , -height/2+sum(thick(1:k))];
   end
else
    hcoord = thick(1:numb_lay,1:2);
    height = -hcoord(1,1)+hcoord(numb_lay,2);
end

% ---------------------------------------------------------------------
% here we go...

% calculate Q_bar matrix for every layer and store them in 3D array
% then, multiply with height coordinates
for k = 1:numb_lay
   E1=prop(k,1); E2=prop(k,2); G12=prop(k,3);
   nu12 = prop(k,4);
   nu21 = E2*nu12/E1;
   a=ang(k);

   m = cos(a);
   n = sin(a);

   Q = [E1/(1-nu12*nu21),     nu21*E1/(1-nu12*nu21), 0;
      nu12*E2/(1-nu12*nu21), E2/(1-nu12*nu21),      0;
      0,                     0,                     G12];

   T = [m^2, n^2 2*m*n;
      n^2, m^2, -2*m*n;
      -m*n, m*n, m^2-n^2];

   Tq = [m^2, n^2, m*n;
      n^2, m^2, -m*n;
      -2*m*n, 2*m*n, m^2-n^2];
```

```
    Q_bar(:,:,k) = inv(T) * Q * Tq;

    A_temp(:,:,k) = Q_bar(:,:,k) * (hcoord(k,2)   - hcoord(k,1)  );
    B_temp(:,:,k) = Q_bar(:,:,k) * (hcoord(k,2)^2 - hcoord(k,1)^2);
    D_temp(:,:,k) = Q_bar(:,:,k) * (hcoord(k,2)^3 - hcoord(k,1)^3);
end

% now get the ABD- matrix by simply summing up the elemente of the
% temporary matrices along the 3rd dimension
A =   1   * sum(A_temp,3);
B = (1/2) * sum(B_temp,3);
D = (1/3) * sum(D_temp,3);

ABD = [A,B;B,D];
abcd = inv(ABD);

% finally, calculate the elastic constants
Ex = inv(height*abcd(1,1));
Ey = inv(height*abcd(2,2));
Gxy = inv(height*abcd(3,3));

nuxy = -(abcd(2,1)/abcd(1,1));
nuyx = -(abcd(1,2)/abcd(2,2));

eta13 = (abcd(3,1)/abcd(1,1));
eta31 = (abcd(1,3)/abcd(3,3));
eta23 = (abcd(3,2)/abcd(2,2));
eta32 = (abcd(2,3)/abcd(3,3));
```

C3. cltc_micro.m

```
function [E1, E2, G12, nu12, nu21] = cltc_micro(matrix, fibre, Vf)
% CLTC_MICRO determines composite properties with law of mixtures
%
% written by: Angelo Garcia
%
% The function expects the elastic properties in the input matrices
% `matrix' and `fibre' in the order [E, G12, nu12].

% ---------------------------------------------------------------------
% calculations; no comment because it's obvious what's going on!

Em = matrix(1); G12m = matrix(2); nu12m = matrix(3);
Ef = fibre(1); G12f = fibre(2); nu12f = fibre(3);

E1 = Ef*Vf + Em*(1-Vf);
E2 = (Ef*Em)/(Ef*(1-Vf)+Em*Vf);

G12 = inv(Vf/G12f + (1-Vf)/G12m);

nu12 = nu12f*Vf + nu12m*(1-Vf);
nu21 = E2*nu12/E1;
```

C4.ctlc_strain.m

```
function [e0x, e0y, g0xy, k0x, k0y, k0xy] = cltc_strain(ABD, load);
% CLTC_STRAIN determines strains at the reference plain
%
% written by: Angelo Garcia

% ---------------------------------------------------------------------
% simple calculations

strains = ABD\load;

% ---------------------------------------------------------------------
```

```
% define variables; output as matrix would be also possible

e0x=strains(1); e0y=strains(2); g0xy=strains(3);
k0x=strains(4); k0y=strains(5); k0xy=strains(6);
```

cltc_sdisp.m

```
function [] = cltc_sdisp(struct,detail)
% CLTC_SDISP displays stuctures with material data
%
% written by: Angelo Garcia

% ---------------------------------------------------------------------
% display

for k=1:size(struct,2)
   fprintf('(%1.0d) ',k)
   fprintf('%c',struct(k).name)
   fprintf('\n')

   if detail == 1
      fprintf(['    Elastic Constants: \n'...
            '    E: %6.4e Pa,    G12: %6.4e Pa,' ...
            '    nu12: %6.4f \n\n'],struct(k).elconst)
   end

   if detail == 2
      fprintf(['    Elastic Constants: \n'...
            '    E: %6.4e Pa,    G12: %6.4e Pa,' ...
            '    nu12: %6.4f \n\n'],struct(k).elconst)
   end

   if detail == 3
      fprintf(['    Elastic Constants: \n'...
            '    E1: %6.4e Pa,    E2: %6.4e Pa \n' ...
            '    G12: %6.4e Pa    nu12: %6.4f \n'] ...
        ,struct(k).elconst)
      fprintf('    Fibre Volume Fraction Vf: %6.4f \n\n' ...
        ,struct(k).vf)
   end

   if detail == 4
      fprintf(['    Elastic Constants: \n'...
            '    Ex: %6.4e Pa,    Ey: %6.4e Pa \n' ...
            '    Gxy: %6.4e Pa    nuxy: %6.4f \n\n'] ...
        ,struct(k).elconst)
      format long g;
      fprintf('    ABD- Matrix: \n')
      disp(struct(k).ABD)
      format short g;
   end
end
```

Appendix D Individual actuators testing results

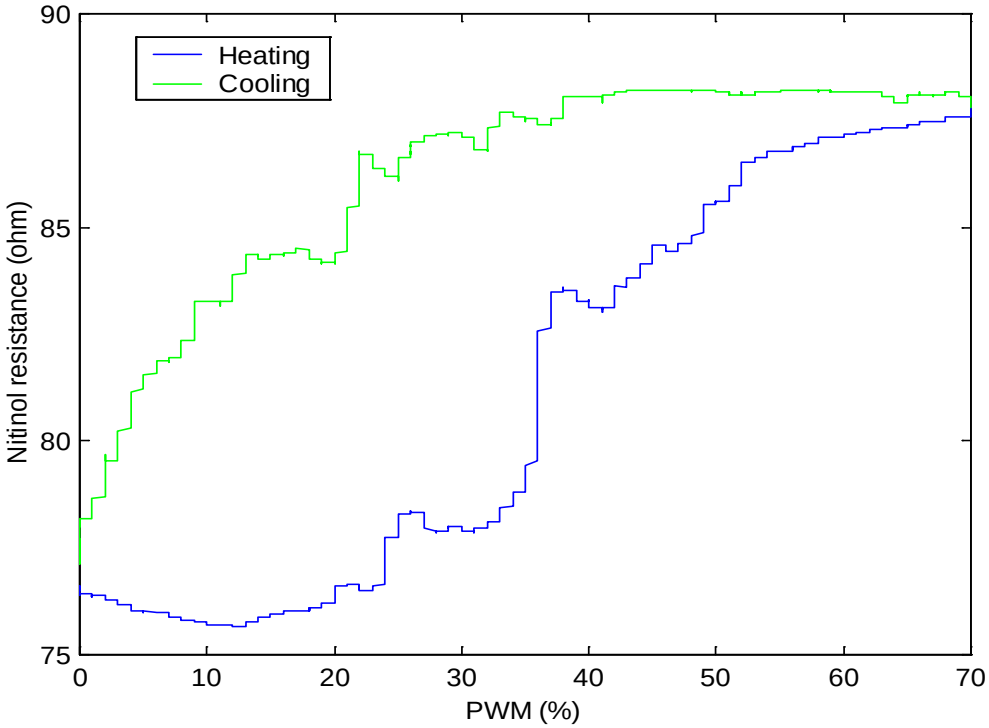


Figure D. 1 The graph of wire resistance-PWM characteristic of the actuator no. 10.

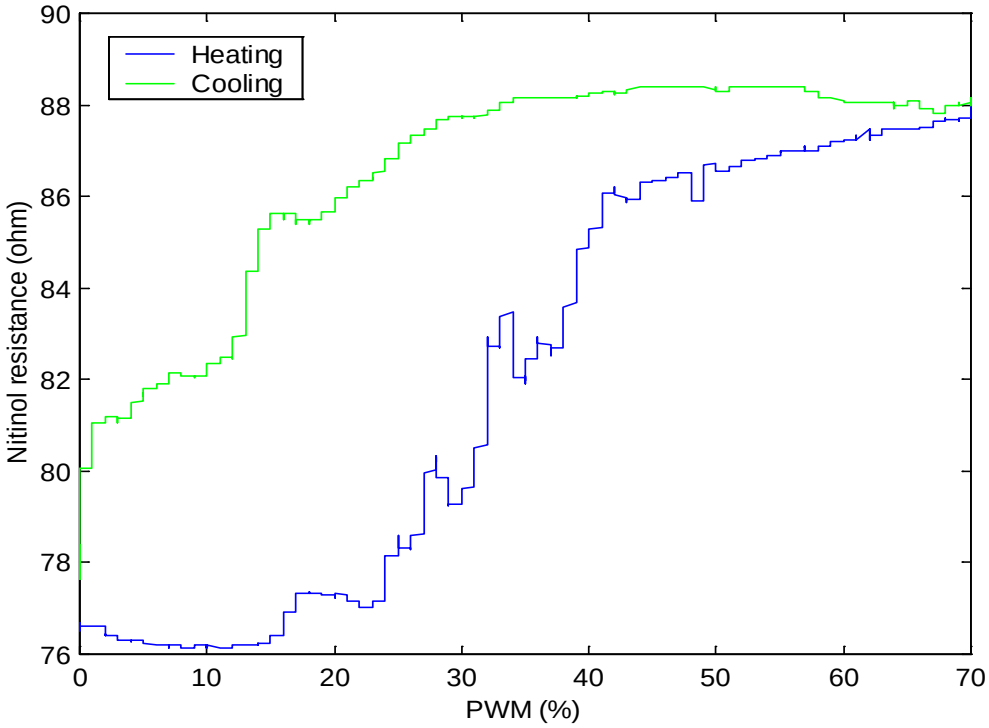


Figure D. 2 The graph of wire resistance-PWM characteristic of the actuator no. 11.

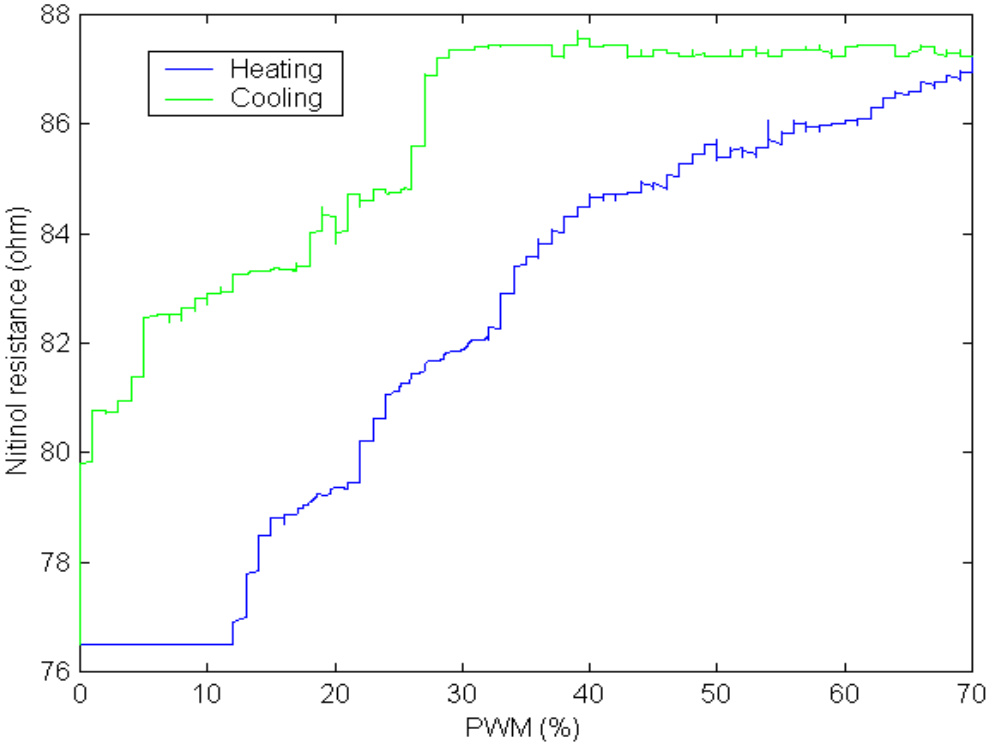


Figure D. 3 The graph of wire resistance-PWM characteristic of the actuator no. 12.

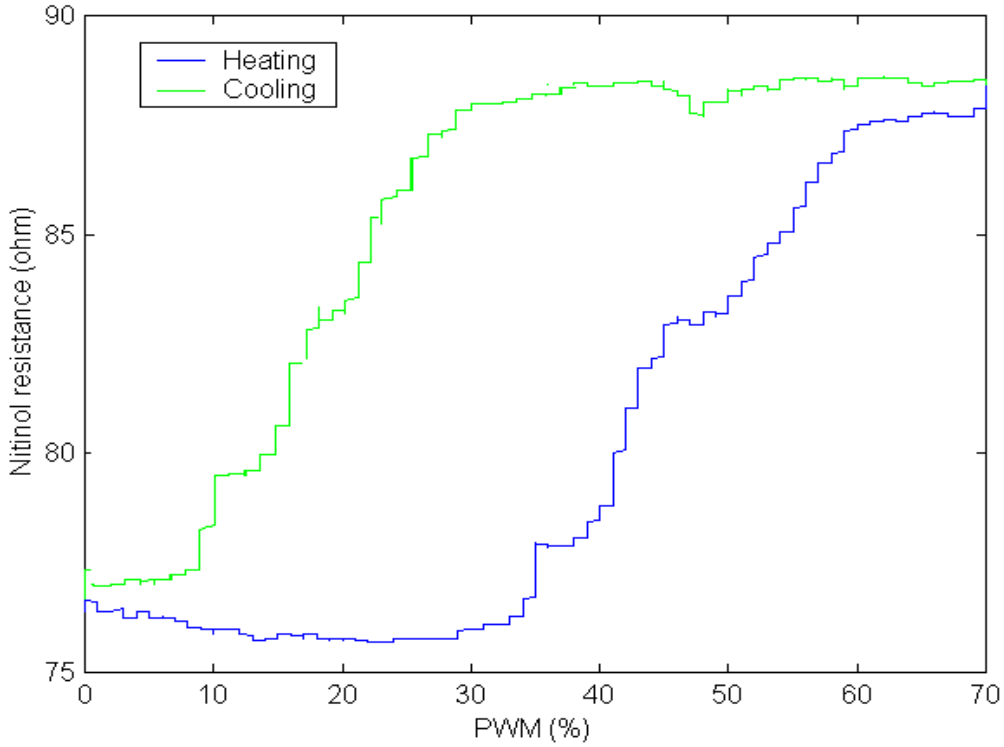


Figure D. 4 The graph of wire resistance-PWM characteristic of the actuator no. 13.

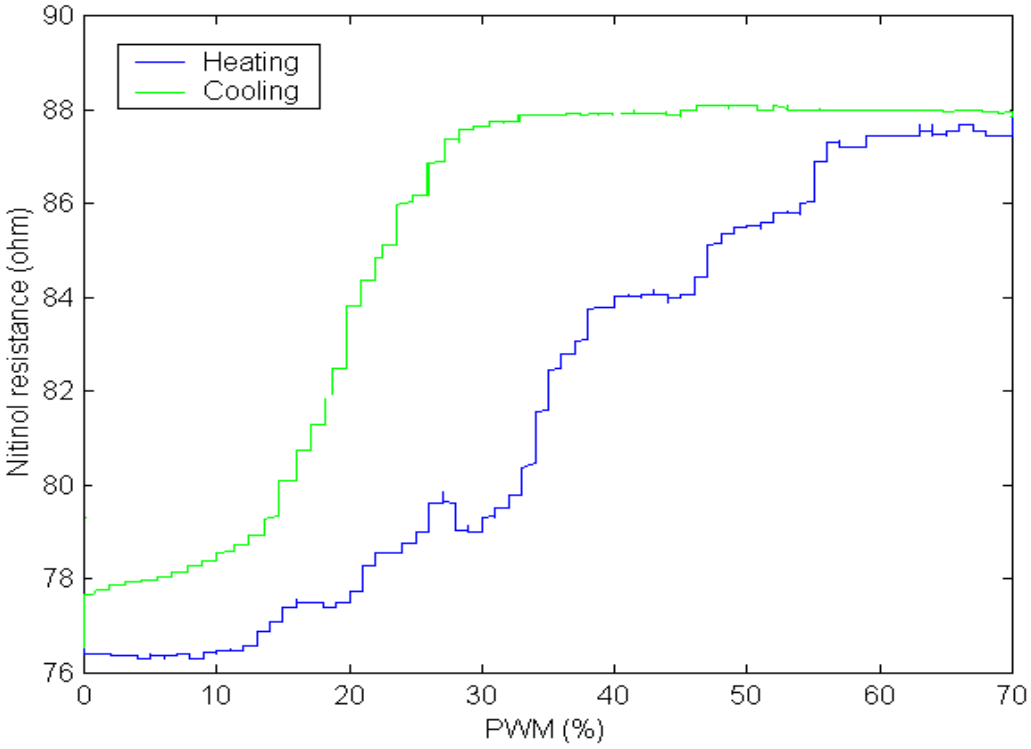


Figure D. 5 The graph of wire resistance-PWM characteristic of the actuator no. 14.

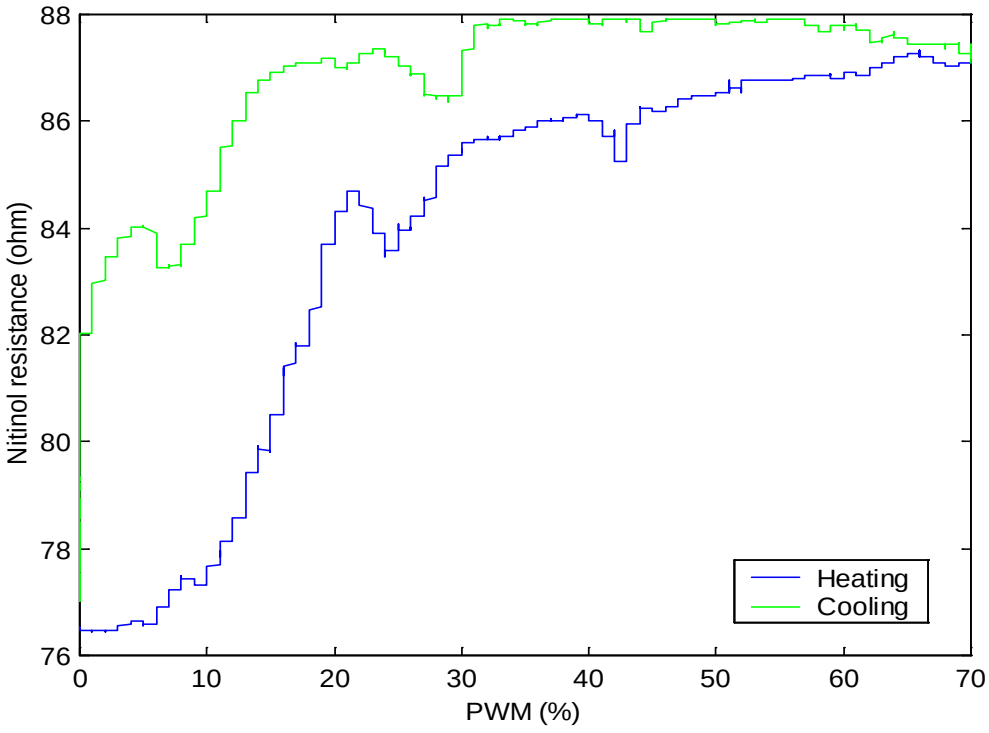


Figure D. 6 The graph of wire resistance-PWM characteristic of the actuator no. 15.

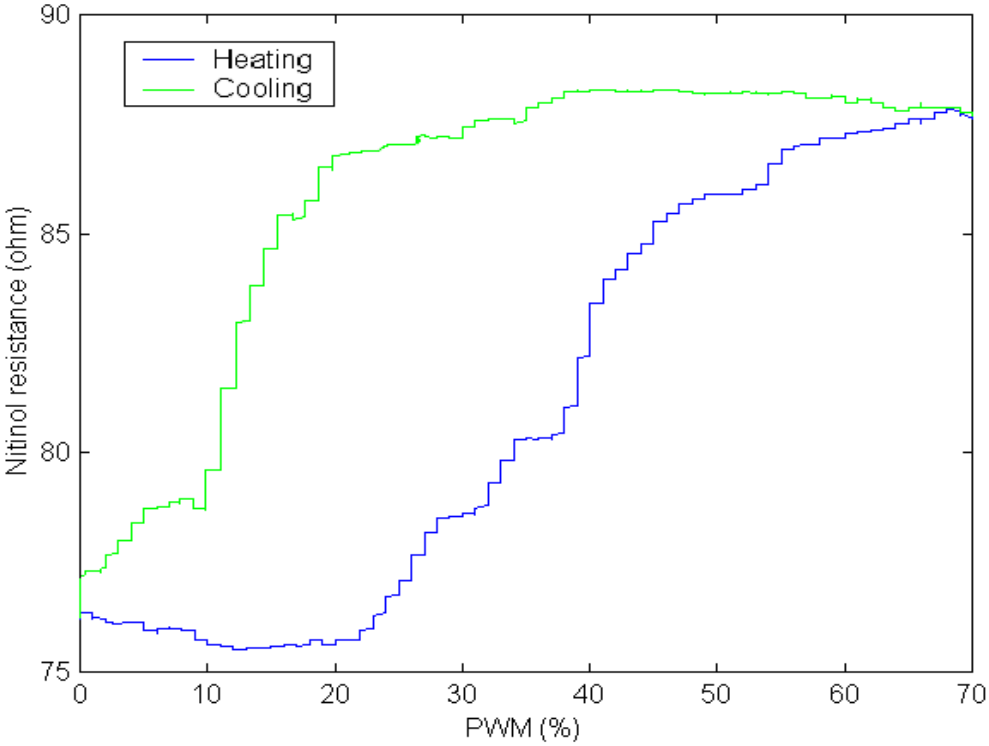


Figure D. 7 The graph of wire resistance-PWM characteristic of the actuator no. 20.

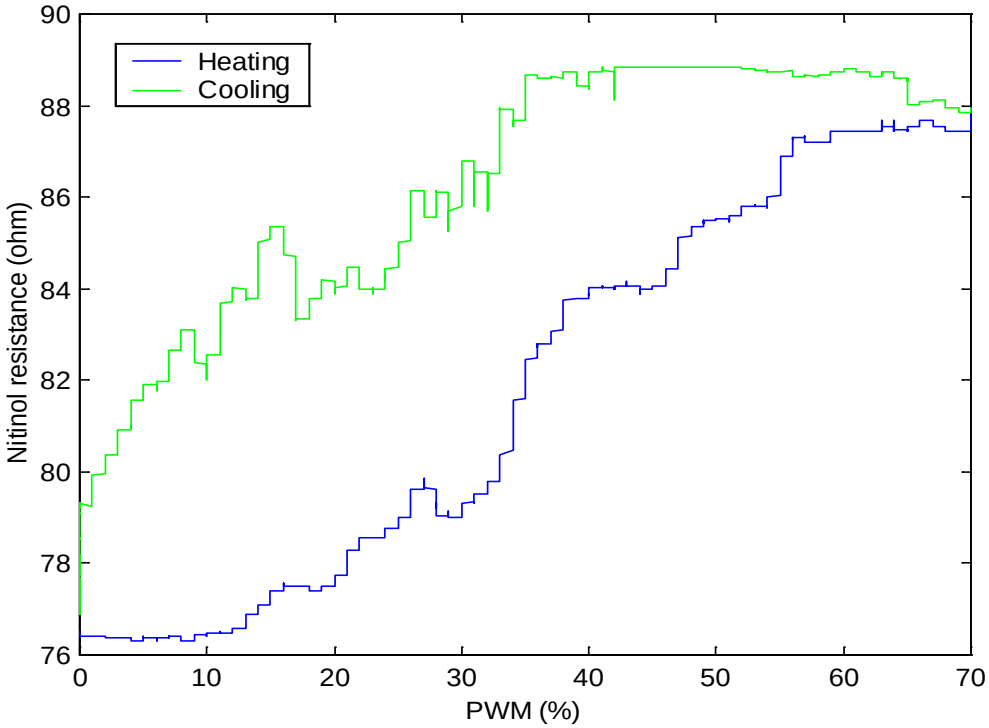


Figure D. 8 The graph of wire resistance-PWM characteristic of the actuator no. 21.

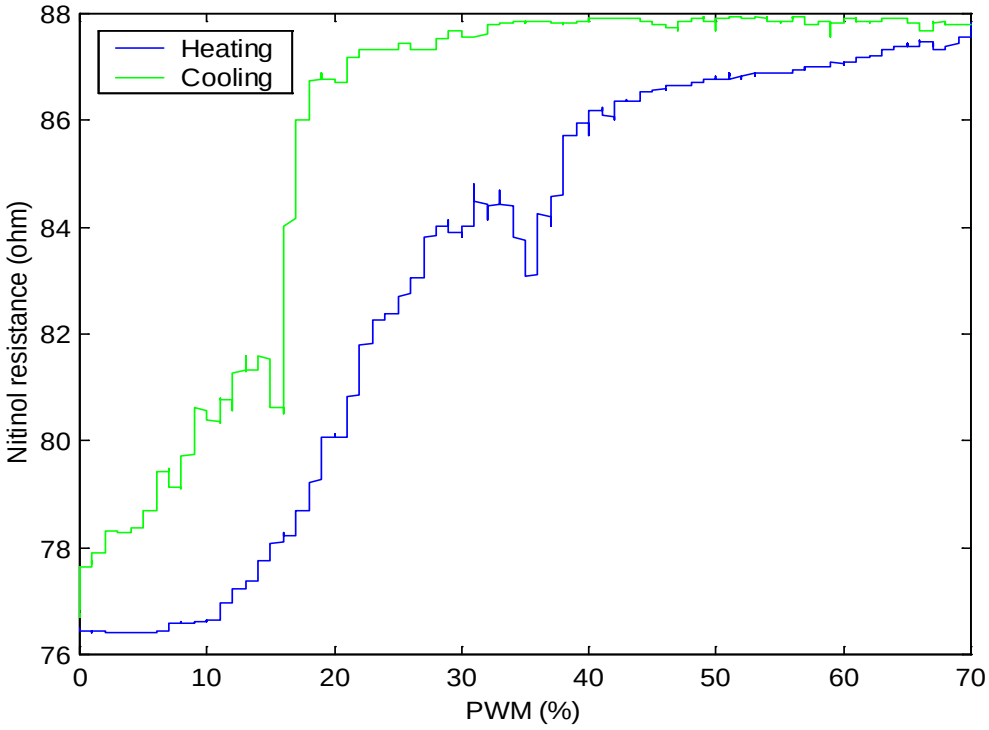


Figure D. 9 The graph of wire resistance-PWM characteristic of the actuator no. 22.

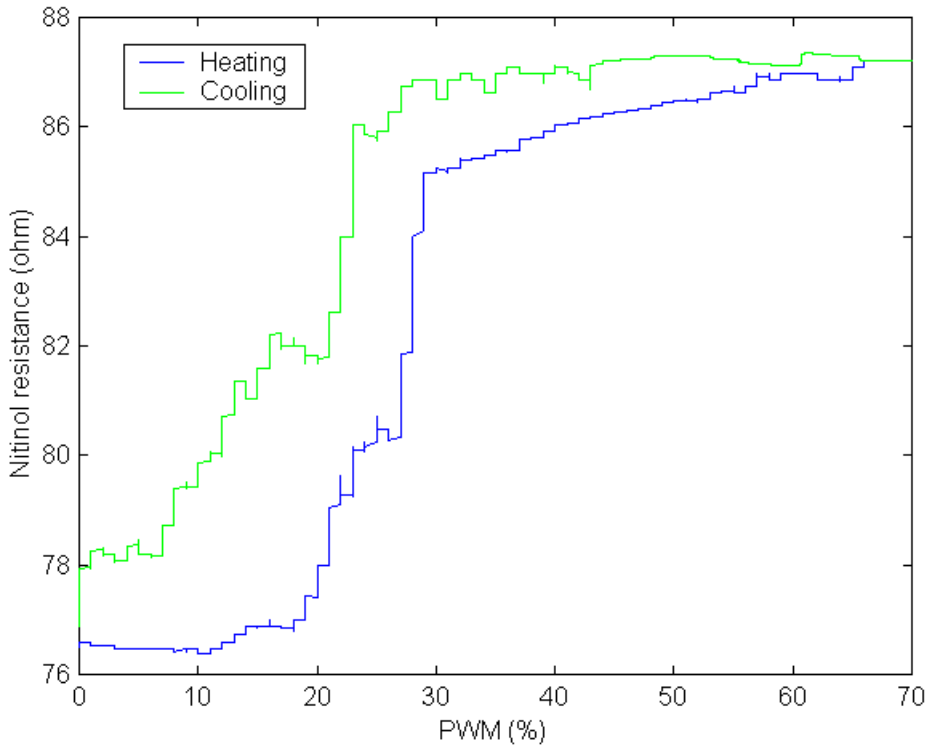


Figure D. 10 The graph of wire resistance-PWM characteristic of the actuator no. 23.

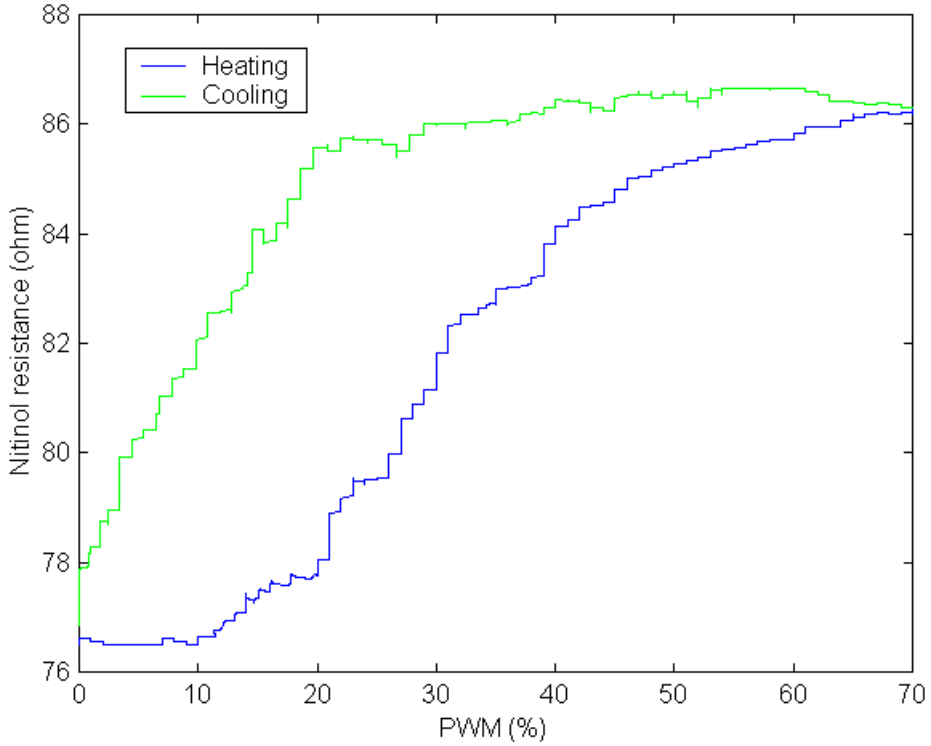


Figure D. 11 The graph of wire resistance-PWM characteristic of the actuator no. 24.

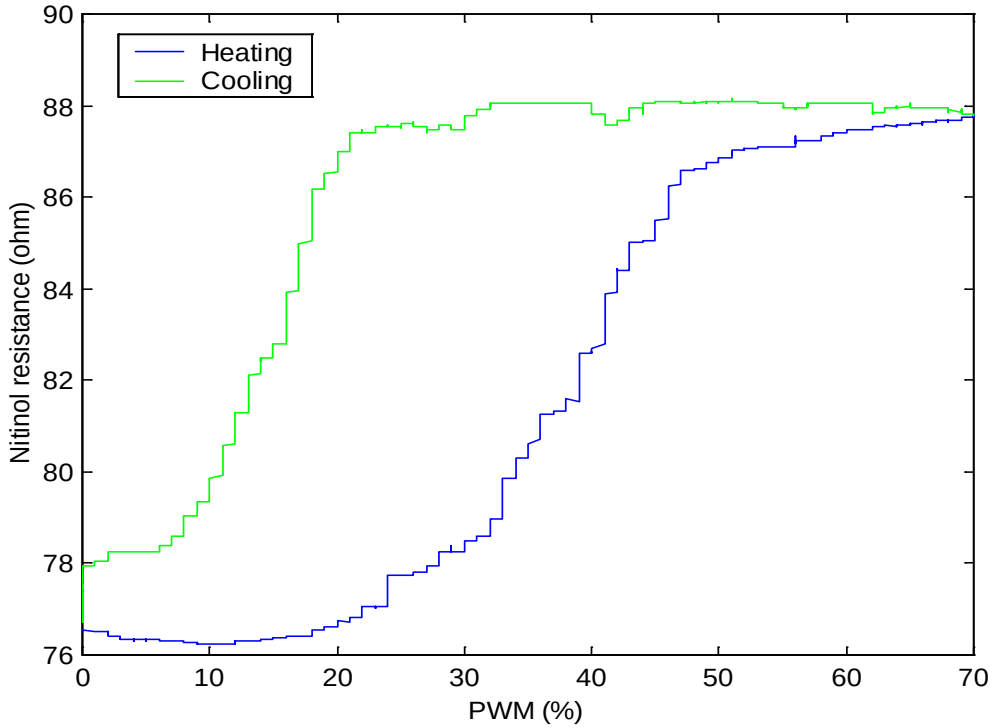


Figure D. 12 The graph of wire resistance-PWM characteristic of the actuator no. 25.

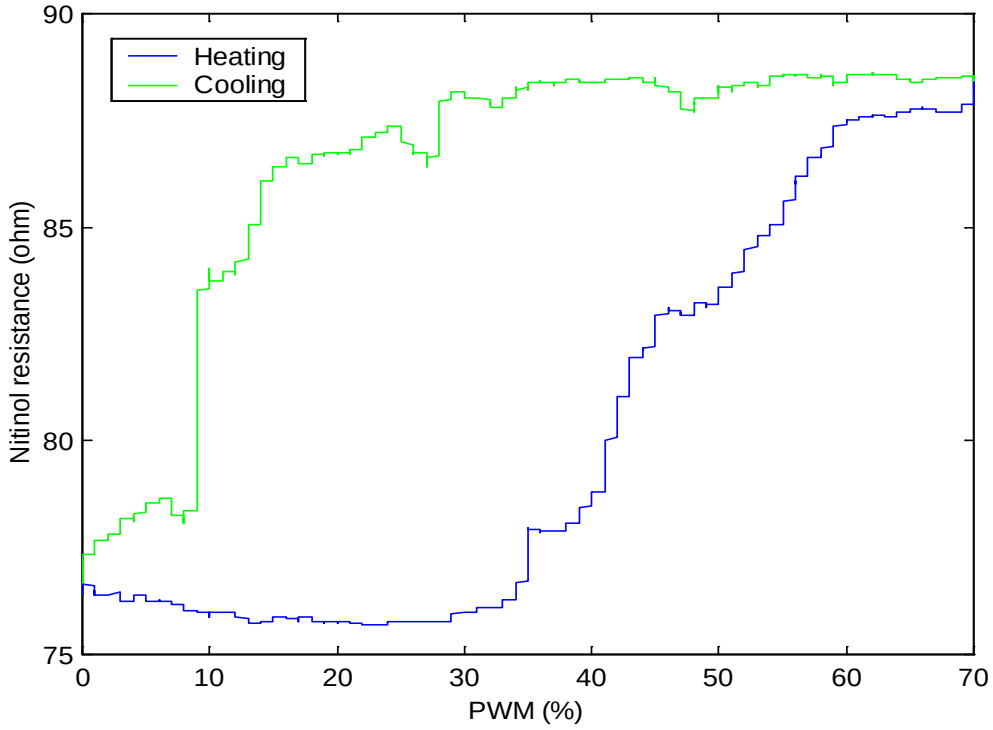


Figure D. 13 The graph of wire resistance-PWM characteristic of the actuator no. 30.

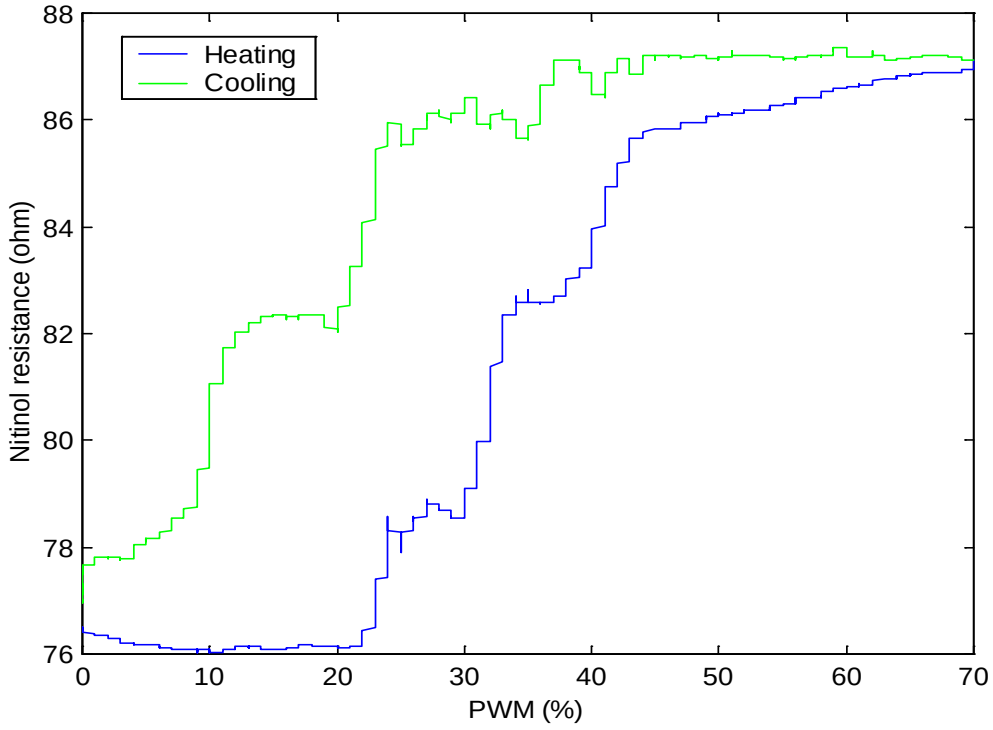


Figure D. 14 The graph of wire resistance-PWM characteristic of the actuator no. 31.

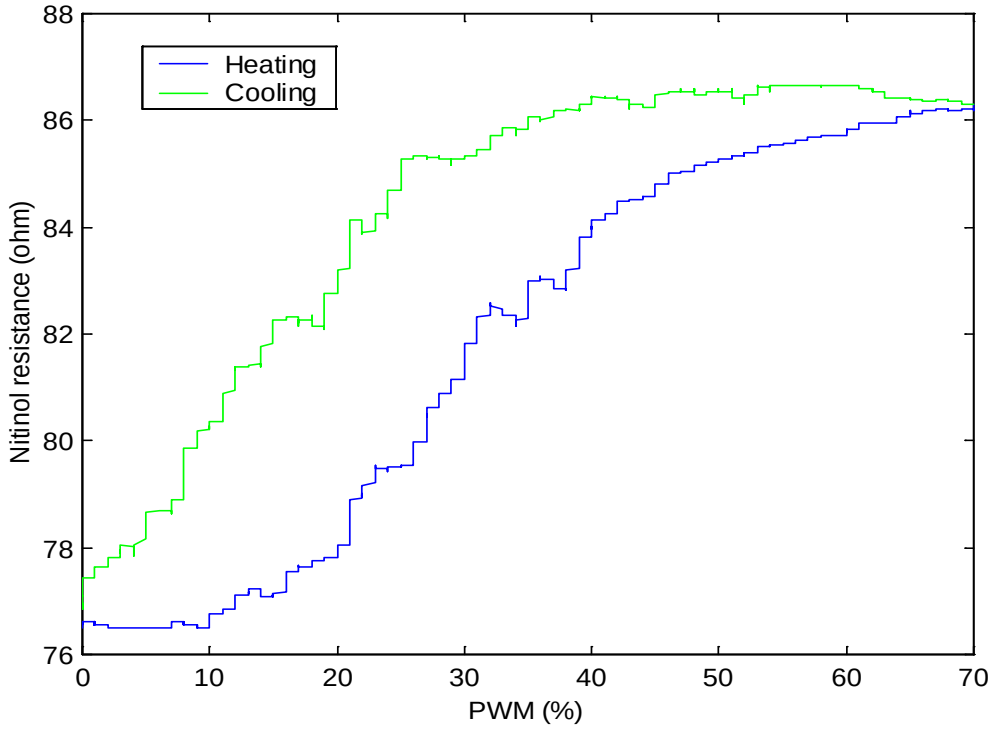


Figure D. 15 The graph of wire resistance-PWM characteristic of the actuator no. 32.

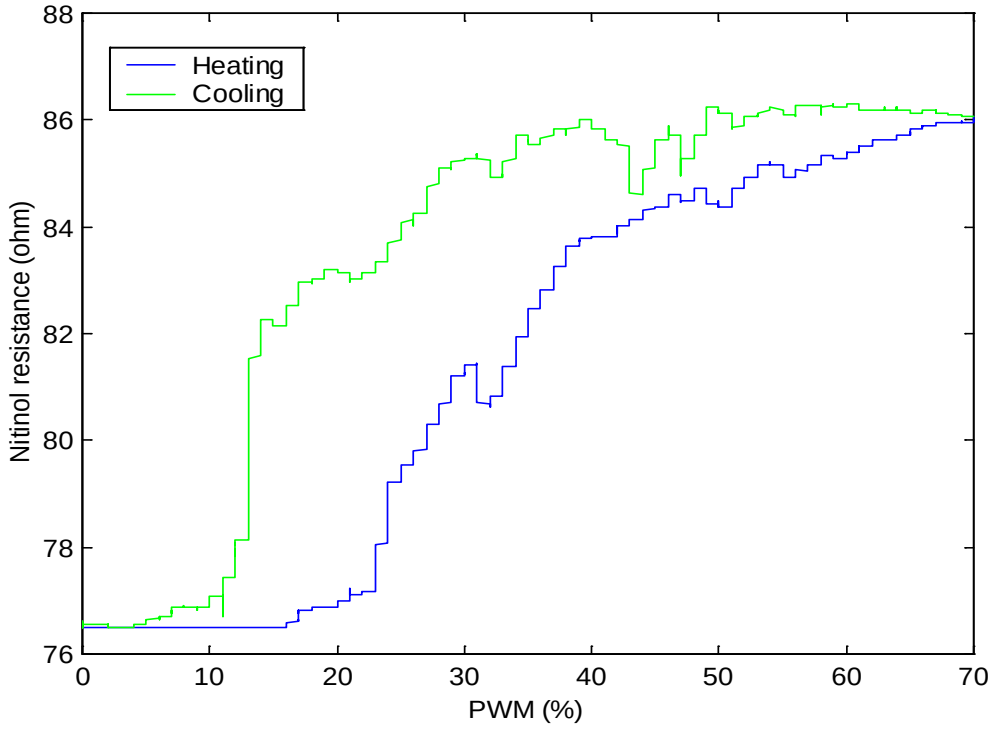


Figure D 16 The graph of wire resistance-PWM characteristic of the actuator no. 33.

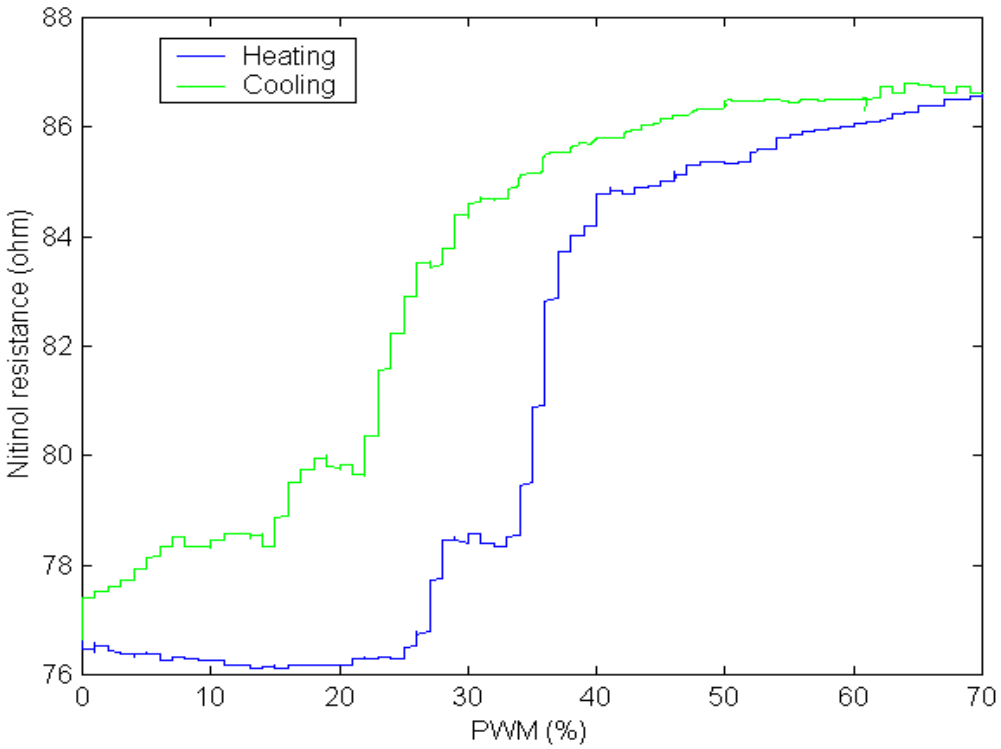


Figure D. 17 The graph of wire resistance-PWM characteristic of the actuator no. 34.

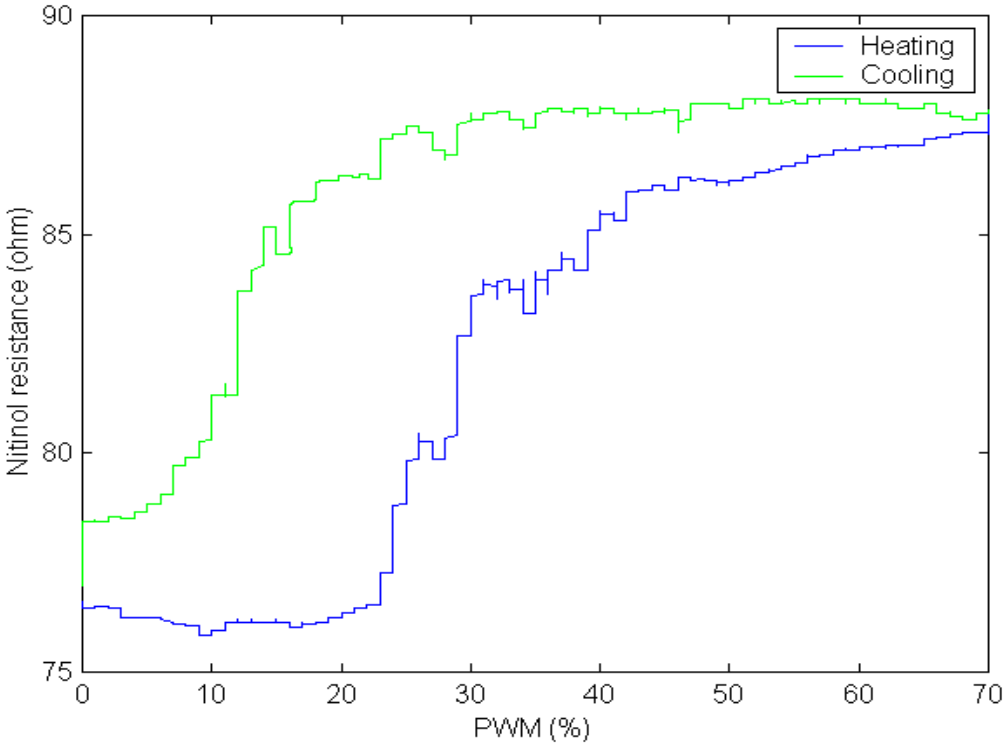


Figure D. 18 The graph of wire resistance-PWM characteristic of the actuator no. 35.

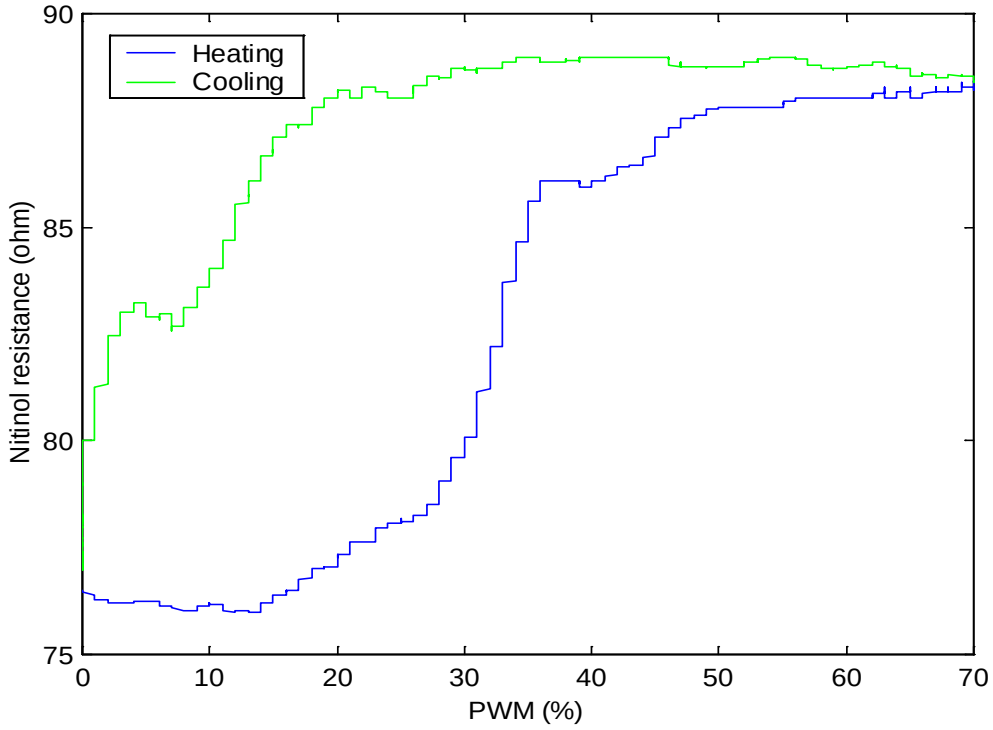


Figure D. 19 The graph of wire resistance-PWM characteristic of the actuator no. 40.

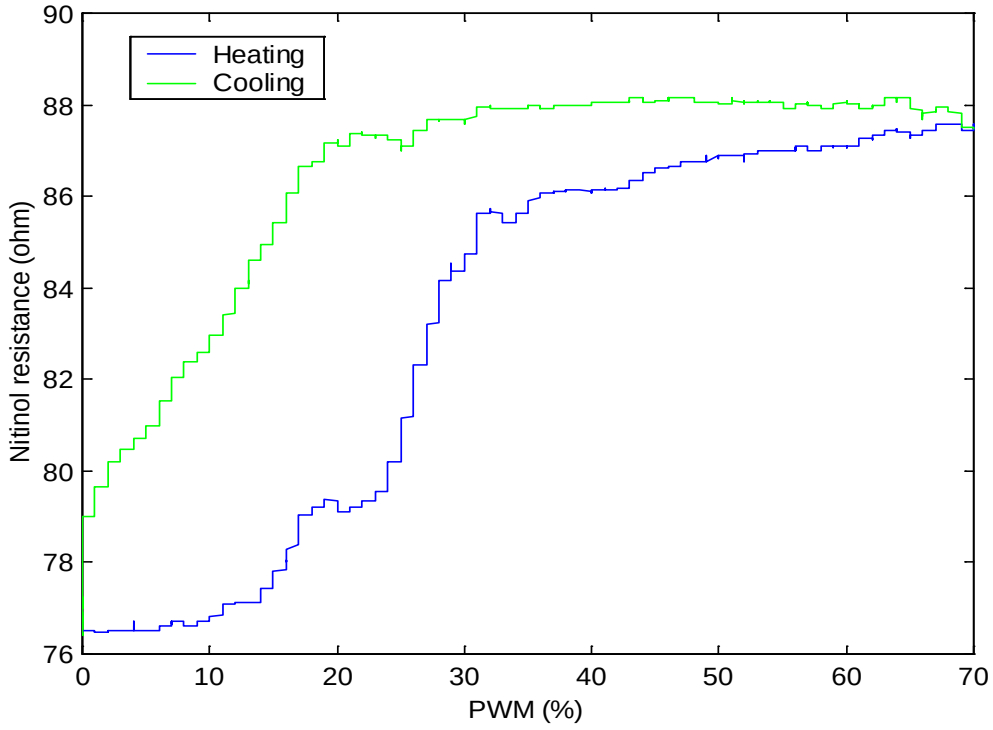


Figure D. 20 The graph of wire resistance-PWM characteristic of the actuator no. 41.

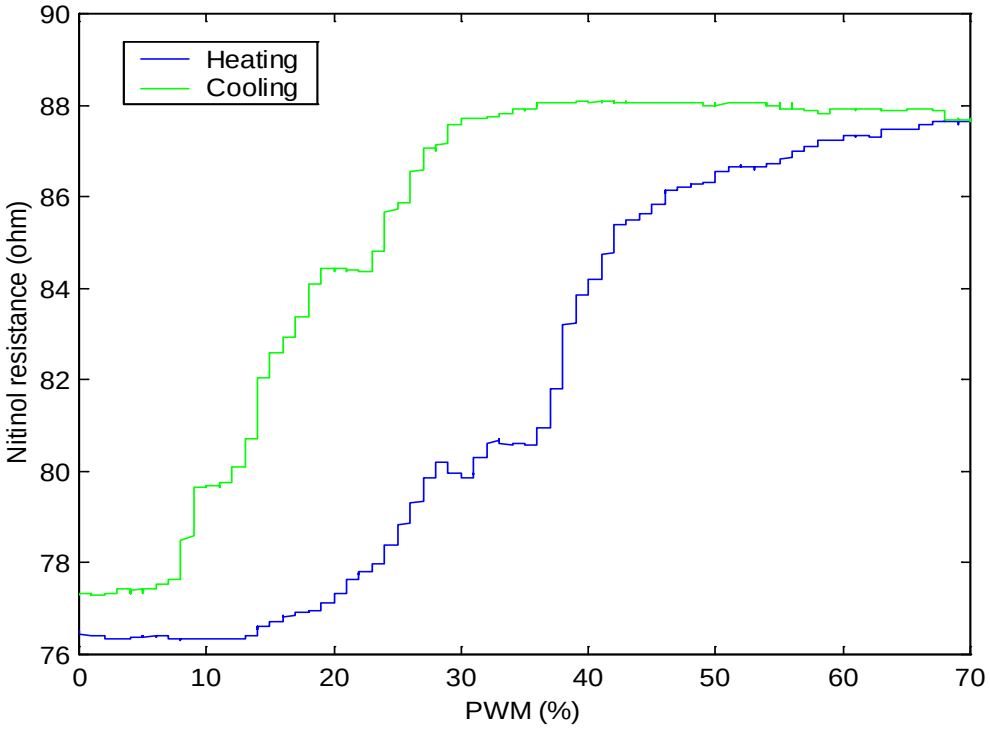


Figure D. 21 The graph of wire resistance-PWM characteristic of the actuator no. 42.

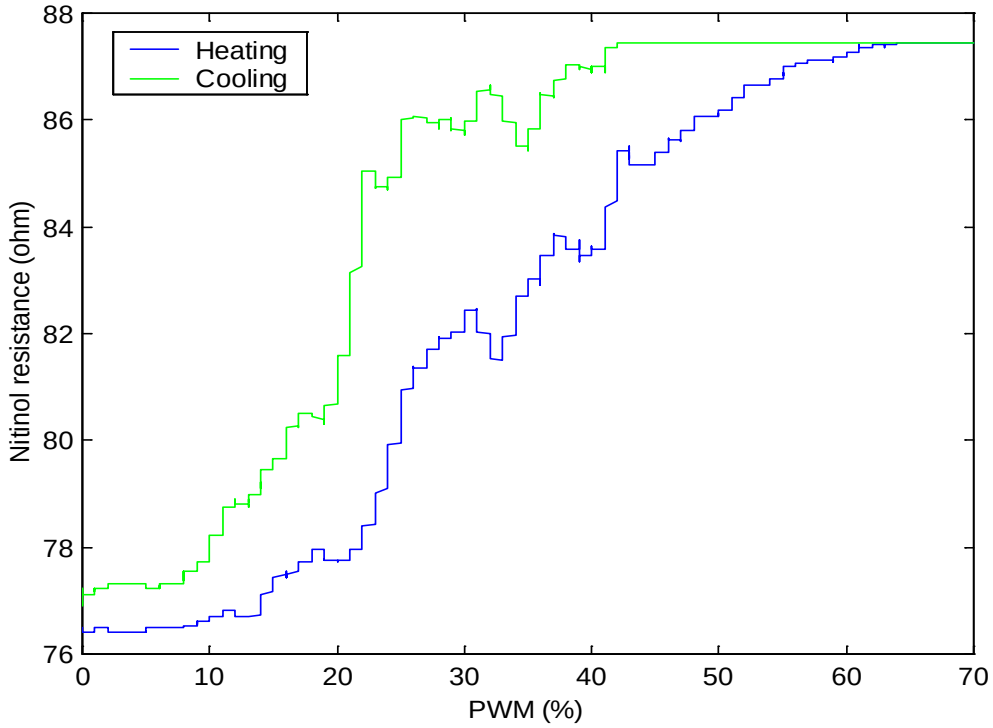


Figure D. 22 The graph of wire resistance-PWM characteristic of the actuator no. 43.

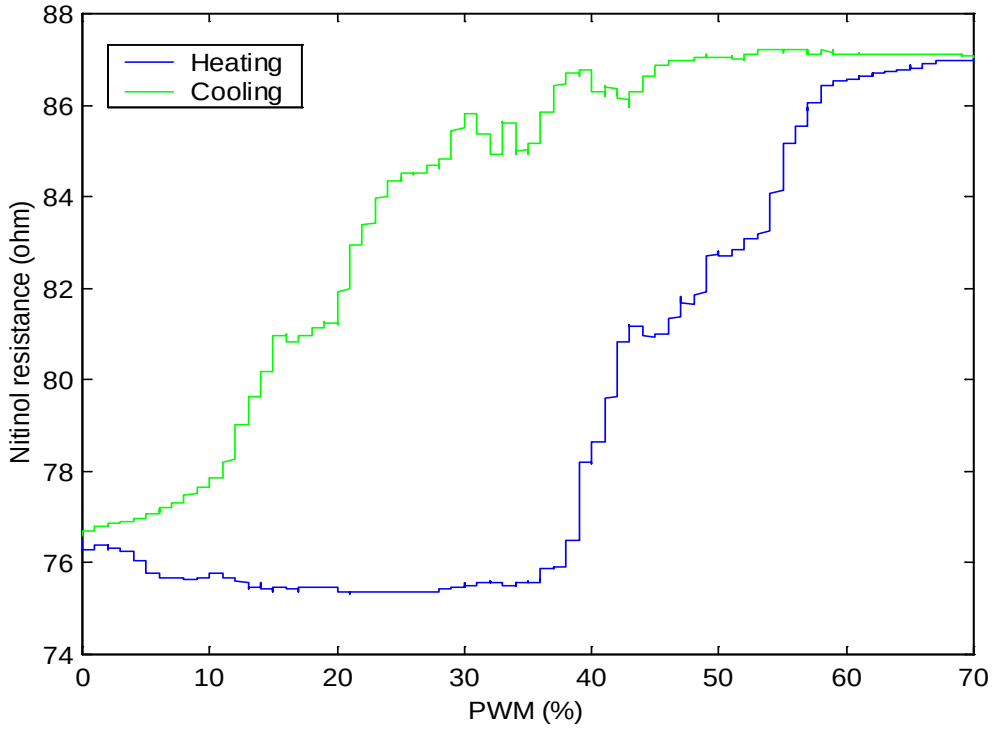


Figure D. 23 The graph of wire resistance-PWM characteristic of the actuator no. 44.

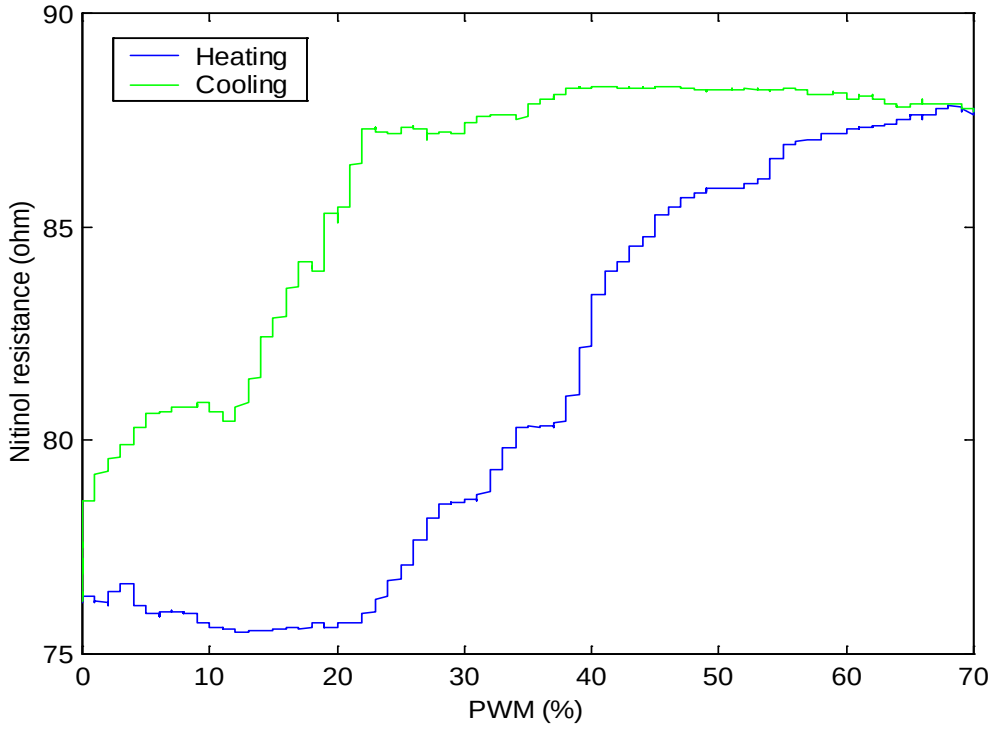


Figure D. 24 The graph of wire resistance-PWM characteristic of the actuator no. 45

Appendix E Technical Properties of Flexinol

This appendix contains the properties of the Ni-Ti wire.

Wire diameter, inch	0.003”
Approximated Resistance, Ω/inch	5
Maximum recommended force, grams	80
Approximated current room temperature, mA	100
Contract time, s	1
Approximated cooling time, s	0.5
Density, lb/in^3	0.235
Specific Heat, BTU/lb°F	0.2
cal(mol.°C)	6-8
Melting Point, °C	1250
Heat of Transformation, BTU/lb	10.4
Thermal Conductivity, cal(cm-°C-sec)	0.05
Thermal Expansion Coefficient	
Martensite, /°C	6.6×10^{-6}
Austenite, /°C	11×10^{-6}
Electrical Resistivity	
Martensite, ohms/cir mil ft	421 (approx)
Austenite, ohms/cir mil ft	511 (approx)
Heat transfer coefficient, $Wm^{-2}°C^{-1}$	7

Appendix F Exact Modelling for Transient Conduction

For an infinite cylinder of r_0 (see Fig F1), which is at an initial uniform temperature and experiences a change in convective conditions. That is, an exact series solution may be obtained for the time dependence of the radial temperature distribution, and a one-term approximation may be used for most conditions. The infinite cylinder is an idealisation that permits the assumption of one-dimensional conduction in the radial direction. It is a reasonable approximation for cylinders having $L/r_0 \geq 10$.

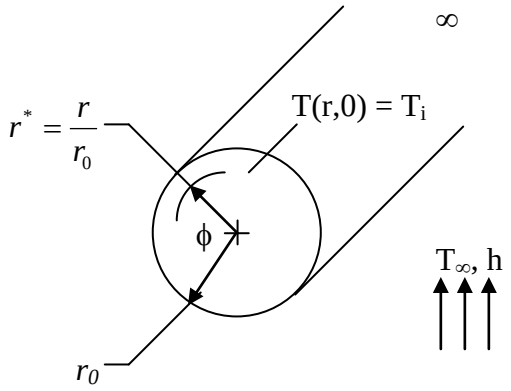


Figure F. 1 Three-dimensional systems with an initial uniform temperature subjected to external heat flux.

Exact solutions to the transient, one-dimensional form of the heat equation have been developed for the infinite cylinder and for the sphere. For a uniform initial temperature and convective boundary conditions, the solutions [17] are as follows.

In one dimensional form, the temperature is

$$\theta^* = \sum C_n \exp\left(-\zeta_n^2 F_0\right) J_0\left(\zeta_n r^*\right) \qquad \text{(F.1)}$$

where $F_0 = \frac{\alpha t}{r_0^2}$,

$$C_n = \frac{2}{\zeta_n} \frac{J_1(\zeta_n)}{J_0^2(\zeta_n) + J_1^2(\zeta_n)} \qquad \text{(F.2)}$$

and the discrete values of ζ_n are positive roots of the transcendental equation

$$B_i = \zeta_n \frac{J_1(\zeta_n)}{J_0(\zeta_n)} \tag{F.3}$$

The quantities J_1 and J_0 are Bessel functions of the first kind and their values are tabulated in Schneider [17]. Roots of the transcendental equation (3) are tabulated by Schneider [17].

Appendix G Control Code

Fsnake.cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#include <stdio.h>
#pragma hdrstop

#include "FSnake.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma link "CPort"
#pragma resource "*.dfm"
TFormSnake *FormSnake;

bool b10,b11,b12,b13,b14,b15;
bool b20,b21,b22,b23,b24,b25;
bool bTop,bBottom,bTime,bAll;

//---------------------------------------------------------------------------
__fastcall TFormSnake::TFormSnake(TComponent* Owner)
  : TForm(Owner)
{
}
//---------------------------------------------------------------------------


void delay(void)
{
  bTime=true;
  FormSnake->Timer->Enabled=true;
  while(bTime);
  FormSnake->Timer->Enabled=false;
}
//---------------------------------------------------------------------------


void __fastcall TFormSnake::FormClose(TObject *Sender,
      TCloseAction &Action)
{
  ComPort->Close();
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::FormShow(TObject *Sender)
{
  ComPort->Open();

  b10=false;
  b11=false;
  b12=false;
  b13=false;
  b14=false;
  b15=false;

  b20=false;
  b21=false;
  b22=false;
  b23=false;
  b24=false;
  b25=false;

  bTop=false;
  bBottom=false;
  bAll=false;
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonALLDownClick(TObject *Sender)
{

  if(bAll)
  {
```

```
    ComPort->WriteStr('m');
    bAll=false;
    PanelLeg10->Color=clNavy;
    PanelLeg11->Color=clNavy;
    PanelLeg12->Color=clNavy;
    PanelLeg13->Color=clNavy;
    PanelLeg14->Color=clNavy;
    PanelLeg15->Color=clNavy;
    PanelLeg20->Color=clNavy;
    PanelLeg21->Color=clNavy;
    PanelLeg22->Color=clNavy;
    PanelLeg23->Color=clNavy;
    PanelLeg24->Color=clNavy;
    PanelLeg25->Color=clNavy;
    b10=false;
    b11=false;
    b12=false;
    b13=false;
    b14=false;
    b15=false;
    b20=false;
    b21=false;
    b22=false;
    b23=false;
    b24=false;
    b25=false;
   }
   else
   {
    ComPort->WriteStr('M');
    bAll=true;
    PanelLeg10->Color=clRed;
    PanelLeg11->Color=clRed;
    PanelLeg12->Color=clRed;
    PanelLeg13->Color=clRed;
    PanelLeg14->Color=clRed;
    PanelLeg15->Color=clRed;
    PanelLeg20->Color=clRed;
    PanelLeg21->Color=clRed;
    PanelLeg22->Color=clRed;
    PanelLeg23->Color=clRed;
    PanelLeg24->Color=clRed;
    PanelLeg25->Color=clRed;
    b10=true;
    b11=true;
    b12=true;
    b13=true;
    b14=true;
    b15=true;
    b20=true;
    b21=true;
    b22=true;
    b23=true;
    b24=true;
    b25=true;
   }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonAllUpClick(TObject *Sender)
{
  ComPort->WriteStr('m');

    PanelLeg10->Color=clNavy;
    PanelLeg11->Color=clNavy;
    PanelLeg12->Color=clNavy;
    PanelLeg13->Color=clNavy;
    PanelLeg14->Color=clNavy;
    PanelLeg15->Color=clNavy;
    b10=false;
    b11=false;
    b12=false;
    b13=false;
    b14=false;
    b15=false;
```

```
    PanelLeg20->Color=clNavy;
    PanelLeg21->Color=clNavy;
    PanelLeg22->Color=clNavy;
    PanelLeg23->Color=clNavy;
    PanelLeg24->Color=clNavy;
    PanelLeg25->Color=clNavy;
    b20=false;
    b21=false;
    b22=false;
    b23=false;
    b24=false;
    b25=false;

}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg10Click(TObject *Sender)
{
  if(b10)
  {
    ComPort->WriteStr('a');
    b10=false;
    PanelLeg10->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('A');
    b10=true;
    PanelLeg10->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg11Click(TObject *Sender)
{
  if(b11)
  {
    ComPort->WriteStr('k');
    b11=false;
    PanelLeg11->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('K');
    b11=true;
    PanelLeg11->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg12Click(TObject *Sender)
{
  if(b12)
  {
    ComPort->WriteStr('l');
    b12=false;
    PanelLeg12->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('L');
    b12=true;
    PanelLeg12->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg14Click(TObject *Sender)
{
  if(b14)
  {
    ComPort->WriteStr('e');
    b14=false;
    PanelLeg14->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('E');
    b14=true;
```

```
    PanelLeg14->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg15Click(TObject *Sender)
{
  if(b15)
  {
    ComPort->WriteStr('b');
    b15=false;
    PanelLeg15->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('B');
    b15=true;
    PanelLeg15->Color=clRed;
  }
}
//---------------------------------------------------------------------------

void __fastcall TFormSnake::ButtonLeg13Click(TObject *Sender)
{
  if(b13)
  {
    ComPort->WriteStr('f');
    b13=false;
    PanelLeg13->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('F');
    b13=true;
    PanelLeg13->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg20Click(TObject *Sender)
{
  if(b20)
  {
    ComPort->WriteStr('c');
    b20=false;
    PanelLeg20->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('C');
    b20=true;
    PanelLeg20->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg21Click(TObject *Sender)
{
  if(b21)
  {
    ComPort->WriteStr('i');
    b21=false;
    PanelLeg21->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('I');
    b21=true;
    PanelLeg21->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg22Click(TObject *Sender)
{
  if(b22)
  {
    ComPort->WriteStr('j');
    b22=false;
```

```
    PanelLeg22->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('J');
    b22=true;
    PanelLeg22->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg23Click(TObject *Sender)
{
  if(b23)
  {
    ComPort->WriteStr('g');
    b23=false;
    PanelLeg23->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('G');
    b23=true;
    PanelLeg23->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg24Click(TObject *Sender)
{
  if(b24)
  {
    ComPort->WriteStr('h');
    b24=false;
    PanelLeg24->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('H');
    b24=true;
    PanelLeg24->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLeg25Click(TObject *Sender)
{
  if(b25)
  {
    ComPort->WriteStr('d');
    b25=false;
    PanelLeg25->Color=clNavy;
  }
  else
  {
    ComPort->WriteStr('D');
    b25=true;
    PanelLeg25->Color=clRed;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonTopClick(TObject *Sender)
{
  if(bTop)
  {
    ComPort->WriteStr('o');
    bTop=false;
    PanelLeg20->Color=clNavy;
    PanelLeg21->Color=clNavy;
    PanelLeg22->Color=clNavy;
    PanelLeg23->Color=clNavy;
    PanelLeg24->Color=clNavy;
    PanelLeg25->Color=clNavy;
    b20=false;
    b21=false;
    b22=false;
    b23=false;
    b24=false;
```

```
    b25=false;
   }
   else
   {
    ComPort->WriteStr('O');
    bTop=true;
    PanelLeg20->Color=clRed;
    PanelLeg21->Color=clRed;
    PanelLeg22->Color=clRed;
    PanelLeg23->Color=clRed;
    PanelLeg24->Color=clRed;
    PanelLeg25->Color=clRed;
    b20=true;
    b21=true;
    b22=true;
    b23=true;
    b24=true;
    b25=true;
   }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonLegRelease1XClick(TObject *Sender)
{
    ComPort->WriteStr('n');
    PanelLeg10->Color=clNavy;
    PanelLeg11->Color=clNavy;
    PanelLeg12->Color=clNavy;
    PanelLeg13->Color=clNavy;
    PanelLeg14->Color=clNavy;
    PanelLeg15->Color=clNavy;
    b10=false;
    b11=false;
    b12=false;
    b13=false;
    b14=false;
    b15=false;

}
//---------------------------------------------------------------------------

void __fastcall TFormSnake::ButtonLegRelease2XClick(TObject *Sender)
{
    ComPort->WriteStr('o');
    PanelLeg20->Color=clNavy;
    PanelLeg21->Color=clNavy;
    PanelLeg22->Color=clNavy;
    PanelLeg23->Color=clNavy;
    PanelLeg24->Color=clNavy;
    PanelLeg25->Color=clNavy;
    b20=false;
    b21=false;
    b22=false;
    b23=false;
    b24=false;
    b25=false;
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::TimerTimer(TObject *Sender)
{
  bTime=false;
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::ButtonBottomClick(TObject *Sender)
{
  if(bBottom)
  {
    ComPort->WriteStr('n');
    bBottom=false;
    PanelLeg10->Color=clNavy;
    PanelLeg11->Color=clNavy;
    PanelLeg12->Color=clNavy;
    PanelLeg13->Color=clNavy;
    PanelLeg14->Color=clNavy;
    PanelLeg15->Color=clNavy;
    b10=false;
```

```
    b11=false;
    b12=false;
    b13=false;
    b14=false;
    b15=false;
   }
   else
   {
    ComPort->WriteStr('N');
    bBottom=true;
    PanelLeg10->Color=clRed;
    PanelLeg11->Color=clRed;
    PanelLeg12->Color=clRed;
    PanelLeg13->Color=clRed;
    PanelLeg14->Color=clRed;
    PanelLeg15->Color=clRed;
    b10=true;
    b11=true;
    b12=true;
    b13=true;
    b14=true;
    b15=true;
   }
}
//---------------------------------------------------------------------------
void __fastcall TFormSnake::Button1Click(TObject *Sender)
{
  ComPort->WriteStr('X');
}
//---------------------------------------------------------------------------

void __fastcall TFormSnake::Button2Click(TObject *Sender)
{
  ComPort->WriteStr('Z');
}
//---------------------------------------------------------------------------
```

Main.c

```
////////////////////////////////////////////////////
//ProjectName..Snake
//Author.......Angelo Garcia
//Purpose......Test operation of Snake hardware.
//History......03-07-2003 Program creation, heartbeat flasher working
//                                      15-07-2003 Verified that all static pins have are working.
//
//
//Serial Parameters
//Baud                  9600
//Data                  8
//Parity      None
//Stop                  1
//Flow                  None
////////////////////////////////////////////////////

#include <stdlib.h>
#include <string.h>
#include <m8c.h>
#include <counter8.h>
#include <uart.h>
#include <PWM8A.h>
#include <PWM8B.h>
#include <PWM8C.h>

#define NAK_RESPONSE    0x00

void HeartBeatL(void);
void HeartBeatH(void);
void HBInit(void);
void HBTick(void);
void HBWaste(void);
void InitPorts(void);
void ConfigComms(void);
void StartupMessage(void);
```

```
void SerialPrint(char *StringIn);

void ConfigComms(void);
void StartupMessage(void);

void LegB1L(void);
void LegB1H(void);
void LegB2L(void);
void LegB2H(void);
void LegB3L(void);
void LegB3H(void);
void LegB4L(void);
void LegB4H(void);
void LegC1L(void);
void LegC1H(void);
void LegC2L(void);
void LegC2H(void);
void LegC3L(void);
void LegC3H(void);
void LegC4L(void);
void LegC4H(void);
void LegA1L(void);
void LegA1H(void);
void LegA2L(void);
void LegA2H(void);
void LegA3L(void);
void LegA3H(void);
void LegA4L(void);
void LegA4H(void);
void LegEnableAL(void);
void LegEnableAH(void);
void LegEnableBL(void);
void LegEnableBH(void);
void LegEnableCL(void);
void LegEnableCH(void);

void GetComms(void);
void ComCommand(void);

int		iHBCount;
long	ltime;
char	cSingleData;

void StartupMessage(void)
{
		while (!(bUART_ReadTxStatus() & UART_TX_BUFFER_EMPTY)) {}
		UART_SendData('S');
		while (!(bUART_ReadTxStatus() & UART_TX_BUFFER_EMPTY)) {}
		UART_SendData('n');
		while (!(bUART_ReadTxStatus() & UART_TX_BUFFER_EMPTY)) {}
		UART_SendData('a');
		while (!(bUART_ReadTxStatus() & UART_TX_BUFFER_EMPTY)) {}
		UART_SendData('k');
		while (!(bUART_ReadTxStatus() & UART_TX_BUFFER_EMPTY)) {}
		UART_SendData('e');
		while (!(bUART_ReadTxStatus() & UART_TX_BUFFER_EMPTY)) {}
		UART_SendData(0x0a);
		while (!(bUART_ReadTxStatus() & UART_TX_BUFFER_EMPTY)) {}
		UART_SendData(0x0d);
}//StartupMessage()


void main()
{
		InitPorts();
		HBInit();
		ConfigComms();
		//Send some data out the serial port
		StartupMessage();

		//Put out some PWM
		PWM8A_DisableInt();
		PWM8B_DisableInt();
		PWM8C_DisableInt();
```

```
PWM8A_WritePeriod(250);
PWM8A_WritePulseWidth(245);
PWM8B_WritePeriod(250);
PWM8B_WritePulseWidth(245);
PWM8C_WritePeriod(250);
PWM8C_WritePulseWidth(245);
PWM8A_Start();
PWM8B_Start();
PWM8C_Start();

LegEnableAH();
LegEnableBH();
LegEnableCH();

LegA1L();
LegA2L();
LegA3L();
LegA4L();

LegB1L();
LegB2L();
LegB3L();
LegB4L();

LegC1L();
LegC2L();
LegC3L();
LegC4L();
//Flash the LED

//GetComms();
//if(cSingleData=='Z')
if(0)
{
	while(1)
	{
		GetComms();
		ComCommand();
	}
}
else
{

ltime=0;
while(1)
{
	switch(ltime)
	{
		//Sequence 1
		case 0:
			LegA2H();
			LegA4H();
		break;
		case 250:
			LegA1L();
			LegA3L();
		break;
		case 500:
			LegB1H();
			LegB4H();
		break;
		case 750:
			LegA2L();
			LegA4L();
		break;
		case 1000:
			LegB2H();
			LegB3H();
		break;
		case 1250:
			LegB1L();
			LegB4L();
		break;
		case 1500:
			LegC4H();
```

```
                LegC2H();
break;
case 1750:
                LegB2L();
                LegB3L();
break;
case 2000:
                LegC3H();
                LegC1H();
break;
case 2250:
                LegC4L();
                LegC2L();
break;
case 2500:
                LegA1H();
                LegA3H();
break;
case 2750:
                LegC3L();
                LegC1L();
break;
////////////////////////////////////////////////
case 3000:
                LegA2H();
                LegA4H();
break;
case 3250:
                LegA1L();
                LegA3L();
break;
case 3500:
                LegB1H();
                LegB4H();
break;
case 3750:
                LegA2L();
                LegA4L();
break;
case 4000:
                LegB2H();
                LegB3H();
break;
case 4250:
                LegB1L();
                LegB4L();
break;
case 4500:
                LegC4H();
                LegC2H();
break;
case 4750:
                LegB2L();
                LegB3L();
break;
case 5000:
                LegC3H();
                LegC1H();
break;
case 5250:
                LegC4L();
                LegC2L();
break;
case 5500:
                LegA1H();
                LegA3H();
break;
case 5750:
                LegC3L();
                LegC1L();
break;
//Sequence 2
case 6000:
                LegA1L();
                LegA2L();
```

```
                LegA3L();
                LegA4L();
                LegB1L();
                LegB2L();
                LegB3L();
                LegB4L();
                LegC1L();
                LegC2L();
                LegC3L();
                LegC4L();
        break;
        case 6250:
                LegA1H();
                LegA2H();
                LegA3H();
                LegA4H();
                LegB1H();
                LegB2H();
                LegB3H();
                LegB4H();
                LegC1H();
                LegC2H();
                LegC3H();
                LegC4H();
        break;
        case 6500:
                LegA1L();
                LegA2L();
                LegA3L();
                LegA4L();
                LegB1L();
                LegB2L();
                LegB3L();
                LegB4L();
                LegC1L();
                LegC2L();
                LegC3L();
                LegC4L();
        break;
        case 6750:
                LegA1H();
                LegA2H();
                LegA3H();
                LegA4H();
                LegB1H();
                LegB2H();
                LegB3H();
                LegB4H();
                LegC1H();
                LegC2H();
                LegC3H();
                LegC4H();
        break;
        case 7000:
                LegA1L();
                LegA2L();
                LegA3L();
                LegA4L();
                LegB1L();
                LegB2L();
                LegB3L();
                LegB4L();
                LegC1L();
                LegC2L();
                LegC3L();
                LegC4L();
        break;
        case 7250:
                LegA1L();
                LegA2L();
                LegA3L();
                LegA4L();
                LegB1L();
                LegB2L();
                LegB3L();
```

```
                LegB4L();
                LegC1L();
                LegC2L();
                LegC3L();
                LegC4L();
        break;
        case 7500:
                LegA1H();
                LegA2H();
                LegA3H();
                LegA4H();
                LegB1H();
                LegB2H();
                LegB3H();
                LegB4H();
                LegC1H();
                LegC2H();
                LegC3H();
                LegC4H();
        break;
        case 7750:
                LegA1L();
                LegA2L();
                LegA3L();
                LegA4L();
                LegB1L();
                LegB2L();
                LegB3L();
                LegB4L();
                LegC1L();
                LegC2L();
                LegC3L();
                LegC4L();
        break;
        //Sequence 3
        case 8000:
                LegC1L();
                LegC2L();
                LegC3L();
                LegC4L();
                LegA1H();
                LegA2H();
                LegA3H();
                LegA4H();
        break;
        case 8250:
                LegA1L();
                LegA2L();
                LegA3L();
                LegA4L();
                LegB1H();
                LegB2H();
                LegB3H();
                LegB4H();
        break;
        case 8500:
                LegB1L();
                LegB2L();
                LegB3L();
                LegB4L();
                LegC1H();
                LegC2H();
                LegC3H();
                LegC4H();
        break;
        case 8750:
                LegC1L();
                LegC2L();
                LegC3L();
                LegC4L();
                LegA1H();
                LegA2H();
                LegA3H();
                LegA4H();
        break;
```

```
                                        case 9000:
                                                LegA1L();
                                                LegA2L();
                                                LegA3L();
                                                LegA4L();
                                                LegB1H();
                                                LegB2H();
                                                LegB3H();
                                                LegB4H();
                                        break;
                                        case 9250:
                                                LegB1L();
                                                LegB2L();
                                                LegB3L();
                                                LegB4L();
                                                LegC1H();
                                                LegC2H();
                                                LegC3H();
                                                LegC4H();
                                        break;
                                        case 9500:
                                                LegA1L();
                                                LegA2L();
                                                LegA3L();
                                                LegA4L();
                                                LegB1L();
                                                LegB2L();
                                                LegB3L();
                                                LegB4L();
                                                LegC1L();
                                                LegC2L();
                                                LegC3L();
                                                LegC4L();
                                        break;
                                }
                                HBTick();
                                HBWaste();
                                ltime++;
                                if(ltime>9750)
                                {
                                        ltime=0;
                                }
                        }
                        }//if()
}//main()


void HBWaste(void)
{
        int     idelay;
        for(idelay=0;idelay<1000;idelay++);
}//HBWaste()


//Ports used here
//Port Data Register
//Port Global Select Register
//Port Drive Mode 0 Register
//Port Drive Mode 1 Register
void InitPorts(void)
{
//0=input 1=output

//Turn off all interrupts for now
        PRT0IE=0x00;
        PRT1IE=0x00;
        PRT2IE=0x00;
        PRT3IE=0x00;
        PRT4IE=0x00;
        PRT5IE=0x00;
//Global select registers
//Serial Coms need these
//Global connections
//P1[5],P1[6]<-------Serial comms
//p4[0],P4[2],P4[4]<-PWM
```

```
		PRT0GS=0x00;
		PRT1GS=0x60;
		PRT2GS=0x00;
		PRT3GS=0x00;
		PRT4GS=0x15;
		PRT5GS=0x00;
//Port drive registers
//Set inputs to high Z and outputs to strong drive
		PRT0DM0=0x00;
		PRT1DM0=0x08;
		PRT2DM0=0xff;
		PRT3DM0=0xff;//RESISTIVE PULL DOWN
		PRT4DM0=0xff;
		PRT5DM0=0xff;

		PRT0DM1=0x00;
		PRT1DM1=0x40;
		PRT2DM1=0xff;
		PRT3DM1=0xff;
		PRT4DM1=0x00;
		PRT5DM0=0x00;
//Set ports to known values
		PRT0DR=0xff;
		PRT1DR=0x0f;
		PRT2DR=0xff;
		PRT3DR=0xff;
		PRT4DR=0xff;
		PRT5DR=0xff;
}//InitPorts()


void HBInit(void)
{
		HeartBeatH();
		iHBCount=0;
}//HBInit()


void HBTick(void)
{
		iHBCount++;
		if(iHBCount>10)
		{
			HeartBeatL();
		}
		if(iHBCount>20)
		{
			HeartBeatH();
			iHBCount=0;
		}
}//HBTick()


void HeartBeatH(void)
{
//P1[7]
PRT5DR|=0x08;
}//HeartBeatH()


void HeartBeatL(void)
{
//P1[7]
PRT5DR&=~0x08;
}//HeartBeatL()

//////////////
// B[1..4] //
//////////////
void LegB1L(void)
{
		//P3[0]
		PRT3DR&=~0x01;
}//LegB1L()
void LegB1H(void)
```

```
{
	//P3[0]
	PRT3DR|=0x01;
}//LegB1H()

void LegB2L(void)
{
	//P3[0]
	PRT3DR&=~0x04;
}//LegB2L()
void LegB2H(void)
{
	//P3[0]
	PRT3DR|=0x04;
}//LegB2H()
void LegB3L(void)
{
	//P3[0]
	PRT3DR&=~0x10;
}//LegB3L()
void LegB3H(void)
{
	//P3[0]
	PRT3DR|=0x10;
}//LegB3H()
void LegB4L(void)
{
	//P3[0]
	PRT3DR&=~0x40;
}//LegB4L()
void LegB4H(void)
{
	//P3[0]
	PRT3DR|=0x40;
}//LegB4H()

//////////////
// C[1..4] //
//////////////
void LegC1L(void)
{
	//P2[0]
	PRT2DR&=~0x01;
}//LegC1L()
void LegC1H(void)
{
	//P2[0]
	PRT2DR|=0x01;
}//LegC1H()

void LegC2L(void)
{
	//P2[0]
	PRT2DR&=~0x04;
}//LegC2L()
void LegC2H(void)
{
	//P2[0]
	PRT2DR|=0x04;
}//LegC2H()
void LegC3L(void)
{
	//P2[0]
	PRT2DR&=~0x10;
}//LegC3L()
void LegC3H(void)
{
	//P2[0]
	PRT2DR|=0x10;
}//LegC3H()
void LegC4L(void)
{
	//P2[0]
	PRT2DR&=~0x40;
}//LegC4L()
```

```
void LegC4H(void)
{
		//P2[0]
		PRT2DR|=0x40;
}//LegC4H()

//////////////
// A[1..4] //
//////////////
void LegA1L(void)
{
		//P3[1]
		PRT3DR&=~0x02;
}//LegA1L()
void LegA1H(void)
{
		//P3[1]
		PRT3DR|=0x02;
}//LegA1H()

void LegA2L(void)
{
		//P3[3]
		PRT3DR&=~0x08;
}//LegA2L()
void LegA2H(void)
{
		//P3[3]
		PRT3DR|=0x08;
}//LegA2H()
void LegA3L(void)
{
		//P3[5]
		PRT3DR&=~0x20;
}//LegA3L()
void LegA3H(void)
{
		//P3[5]
		PRT3DR|=0x20;
}//LegA3H()
void LegA4L(void)
{
		//P3[7]
		PRT3DR&=~0x80;
}//LegA4L()
void LegA4H(void)
{
		//P3[7]
		PRT3DR|=0x80;
}//LegA4H()

//////////////////
// Enable[1..3] //
//////////////////
void LegEnableAL(void)
{
		//P4[1]
		PRT4DR&=~0x02;
}//LegEnableAL()
void LegEnableAH(void)
{
		//P4[1]
		PRT4DR|=0x02;
}//LegEnableAH()
void LegEnableBL(void)
{
		//P4[3]
		PRT4DR&=~0x08;
}//LegEnableAL()
void LegEnableBH(void)
{
		//P4[3]
		PRT4DR|=0x08;
}//LegEnableAH()
void LegEnableCL(void)
```

```
{
	//P4[3]
	PRT4DR&=~0x20;
}//LegEnableCL()
void LegEnableCH(void)
{
	//P4[3]
	PRT4DR|=0x20;
}//LegEnableCH()


void GetComms(void)
{
	cSingleData=0x00;
		/* wait for data to be received */
	while( !( bUART_ReadRxStatus() & UART_RX_COMPLETE ) )
	{
			HBTick();
			HBWaste();
		}
		/* preset response to error */
	cSingleData = NAK_RESPONSE;
	/* if no error condition then get byte received */
	if ( ! (bUART_ReadRxStatus() & UART_RX_NO_ERROR) )
	{
	/* no error received */
	cSingleData = bUART_ReadRxData();
	}
	/* transmit the response data */
	UART_SendData(cSingleData);
}//GetComms()


void ComCommand(void)
{
	switch(cSingleData)
	{
	case 'A':
		LegA1H();
	break;
	case 'B':
		LegA2H();
	break;
	case 'C':
		LegA3H();
	break;
	case 'D':
		LegA4H();
	break;
	case 'E':
		LegB1H();
	break;
	case 'F':
		LegB2H();
	break;
	case 'G':
		LegB3H();
	break;
	case 'H':
		LegB4H();
	break;
	case 'I':
		LegC1H();
	break;
	case 'J':
		LegC2H();
	break;
	case 'K':
		LegC3H();
	break;
	case 'L':
		LegC4H();
	break;
	//////////////////////////
	case 'a':
```

```
                LegA1L();
break;
case 'b':
                LegA2L();
break;
case 'c':
                LegA3L();
break;
case 'd':
                LegA4L();
break;
case 'e':
                LegB1L();
break;
case 'f':
                LegB2L();
break;
case 'g':
                LegB3L();
break;
case 'h':
                LegB4L();
break;
case 'i':
                LegC1L();
break;
case 'j':
                LegC2L();
break;
case 'k':
                LegC3L();
break;
case 'l':
                LegC4L();
break;
//Special commands
case 'm':
                                LegA1L();
                                LegA2L();
                                LegA3L();
                                LegA4L();
                                LegB1L();
                                LegB2L();
                                LegB3L();
                                LegB4L();
                                LegC1L();
                                LegC2L();
                                LegC3L();
                                LegC4L();
break;
case 'M':
                                LegA1H();
                                LegA2H();
                                LegA3H();
                                LegA4H();
                                LegB1H();
                                LegB2H();
                                LegB3H();
                                LegB4H();
                                LegC1H();
                                LegC2H();
                                LegC3H();
                                LegC4H();
break;
case 'n':
                                LegA1L();
                                LegA2L();
                                LegB1L();
                                LegB2L();
                                LegC3L();
                                LegC4L();
break;
case 'N':
                                LegA1H();
                                LegA2H();
```

```
                                        LegB1H();
                                        LegB2H();
                                        LegC3H();
                                        LegC4H();
            break;
            case 'o':
                                        LegA3L();
                                        LegA4L();
                                        LegB3L();
                                        LegB4L();
                                        LegC1L();
                                        LegC2L();
            break;
            case 'O':
                                        LegA3H();
                                        LegA4H();
                                        LegB3H();
                                        LegB4H();
                                        LegC1H();
                                        LegC2H();
            break;
            }
}//ComCommand


void ConfigComms(void)
{
            // Set up operation of timer "ClkGEN"
            // The period of 156 divides the 24Mhz clock to 153.8 KHz (Ideal freq = 153.6 kHz)
            Counter8_WritePeriod(156);
   Counter8_DisableInt();
   Counter8_Start();
   //Start up the UART
            UART_Start(UART_PARITY_NONE);
}//ConfigComms()
```

Reference

[1] J. M. Conrad and J. W. Mills, *Stiquito : advanced experiments with a simple and inexpensive robot*. Los Alamitos, CA: IEEE Computer Society Press, 1998.

[2] S. Hirose, *Biologically inspired robots : snake-like locomotors and manipulators*. Oxford ; New York: Oxford University Press, 1993.

[3] T. W. Duerig, *Engineering aspects of shape memory alloys*. London ; Boston: Butterworth-Heinemann, 1990.

[4] K. Ikuta, M. Tsukamoto, and S. Hirose, "Mathematical model and experimental verification of shape memory alloy for designing micro actuator," presented at Proceedings. IEEE Micro Electro Mechanical Systems
Proceedings of the 1991 IEEE Micro Electro Mechanical Systems - MEMS '91,Jan 30-Feb 2 1991, Nara, Jpn, 1991.

[5] D. R. Madill and D. Wang, "Modeling and L2-stability of a shape memory alloy position control system," *IEEE Transactions on Control Systems Technology*, vol. 6, pp. 473-481, 1998.

[6] F. P. Incropera and D. P. DeWitt, *Fundamentals of heat and mass transfer*, 4th ed. New York: Wiley, 1996.

[7] J. P. Merlet, *Parallel robots*. Dordrecht ; Boston, MA: Kluwer Academic Publishers, 2000.

[8] D. A. Stewart, "platform with 6 Degrees of Freedom," *Proc. 1. Mech. E. (Lomdon)*, vol. Vol. 180, 1965-1966.

[9] V. E. Gough and S. G. Whitehall, "Universal tire test machine," *Proc 9th int Technical Congress FISITA*, vol. vol 117, pp. 117-135, 1962.

[10] D. T. Pham, *Techniques, hardware and software for robotic assembly : a thesis presented for the Degree of Doctor of Philosophy in Mechanical Engineering at the University of Canterbury, Christchurch ; New Zealand*, 1979.

[11] G. P. Rathbun, *A Stewart platform six-axis milling machine development : a thesis submitted in fulfilment of the requirements for the degree of Master of Engineering in the University of Canterbury*, 1986.

[12] N. V. Afzulpurkar, *Kinematics, design, programming and control of a robotic platform for satellite tracking and other applications : a thesis presented for the*

degree of Doctor of Philosophy in Mechanical Engineering at the University of Canterbury, Christchurch, New Zealand, 1990.

[13] F. P. Beer, E. R. Johnston, and J. T. DeWolf, *Mechanics of materials*, 3rd ed. New York: McGraw-Hill, 2002.

[14] G. R. Dunlop, "Feedback control of a nitinol wire actuator," presented at Proc. 5th WSES/IEEE World Multiconference on Circuits Systems, Communications & Computers, Rethymnon, Greece, 2001.

[15] H. Tobushi, Y. Ohashi, T. Hori, and H. Yamamoto, "Cyclic deformation of TiNi shape-memory alloy helical spring," *Experimental Mechanics*, vol. 32, pp. 304-308, 1992.

[16] T. M. Hasegawa, S., "A control system to compensate the hysteresis by Preisach model on SMA actuator," presented at Micromechatronics and Human Science, 1998. MHS '98. Proceedings of the 1998 International Symposium on, 1998.

[17] P. J. Schneider, *Conduction heat transfer*. Reading, Mass.: Addison-Wesley, 1957.

Bibliography

[1] L. C. Thomas, *Fundamentals of heat transfer*. Englewood Cliff, N.J.: Prentice-Hall, 1980.

[2] G. E. Myers, *Analytical methods in conduction heat transfer*. N.Y.: McGraw-Hill, 1971.

[3] A. F. Mills, *Heat transfer*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.

[4] S. Middleman, *An introduction to mass and heat transfer : principles of analysis and design*. New York: John Wiley & Sons, 1998.

[5] W. S. Janna and M. Heikal, *Engineering heat transfer*, SI / ed. London: VNR International, 1988.

[6] B. Gebhart, *Heat transfer*, 2d . ed. New York: McGraw-Hill, 1971.

[7] A. J. Chapman, *Heat transfer*, 3d . ed. New York: Macmillan, 1974.

[8] H. S. Carslaw and J. C. Jaeger, *Conduction of heat in solids*, 2d . ed. Oxford: Clarendon Press, 1959.

[9] J. J. Craig, *Introduction to robotics : mechanics and control*, 2nd . ed. Reading, Mass.: Addison-Wesley, 1989.

[10] J. Duffy, *Statics and kinematics with applications to robotics*. Cambridge [England] ; New York, NY, USA: Cambridge University Press, 1996.

[11] M. Fielding, "Omnidirectional gait generating algorithm for hexapod robot : a thesis presented for the degree of Doctor of Philosophy in Mechanical Engineering at the University of Canterbury, Christchurch, New Zealand," 2002, pp. xv, 187.

[12] K. C. Gupta, *Mechanics and control of robots : with 38 illustrations*. New York: Springer, 1997.

[13] J. Lenarcic, M. L. Husty, and International Federation for the Theory of Machines and Mechanisms., *Advances in robot kinematics : analysis and control*. Dordrecht ; Boston: Kluwer Academic Publishers, 1998.

[14] J. Lenarcic and B. Ravani, *Advances in robot kinematics and computational geometry*. Dordrecht ; Boston: Kluwer Academic Publishers, 1994.

[15] D. R. Madill, "Modelling and Stability of a Shape Memory Alloy Position Control System," in *Electrical Engineering*: University of Waterloo, 1993.

[16] E. T. P. Wong, "Use of a Delta robot as a walking machine : a thesis submitted in partial fulfilment of the requirements for the degree of Master of Engineering in the University of Canterbury / by Edward Wong Ting Ping.," 1999.

[17] J. L. C. J. L. C. Yang, "Computer controlled multi-axis robot : a thesis submitted in partial fulfilment of the requirements for the degree of Masters [i.e. Master] of Engineering in the Department of Mechanical Engineering, University of Canterbury, Christchurch, New Zealand / by Jeffery L.C. Yang.," 2000.

[18] G. S. Chirikjian, "Binary paradigm for robotic manipulators," presented at Proceedings of the 1994 IEEE International Conference on Robotics and Automation,May 8-13 1994, San Diego,CA,USA, 1994.

[19] K. J. De Laurentis, A. Fisch, J. Nikitczuk, and C. Mavroidis, "Optimal design of shape memory alloy wire bundle actuators," presented at 2002 IEEE International Conference on Robotics adn Automation,May 11-15 2002, Washington, DC,United States, 2002.

[20] T. Hasegawa and S. Majima, "Control system to compensate the hysteresis by Preisach model on SMA actuator," presented at Proceedings of the 1998 9th International Symposium on Micromechatronics and Human Science, MHS,Nov 25-28 1998, Nagoya,Jpn, 1998.

[21] M. L. Husty, "An algorithm for solving the direct kinematics of general Stewart-Gough Platform," *Mech. Mach. Theory*, vol. v31, pp. 365-380, 1996.

[22] V. I. Itin, V. E. Gyunter, S. A. Shabalovskaya, and R. L. C. Sachdeva, "Mechanical properties and shape memory of porous nitinol," *Materials Characterization*, vol. 32, pp. 179-187, 1994.

[23] T.-Y. Lee and J.-K. Shim, "Algebraic Elimination-Based Real-Time Forward Kinematics of the 6-6 Stewart Platform with Planar Base and Platform," presented at Proceedings of the 2001 IEEE International Conference on Robotics & Automation, 2001.

[24] M.-J. Liu, L. Cong-Xin, and L. Chong-Ni, "Dynamics analysis of the Gough-Stewart Platform Manipulator," *IEEE Transactions on Robotics and Automation*, vol. v16, 2000.

[25] S. Maeda, K. Abe, K. Yamamoto, O. Tohyama, and H. Ito, "Active endoscope with SMA (shape memory alloy) coil springs," presented at Proceedings of the 1995 9th Annual International Workshop on Micro Electro Mechanical Systems,Feb 11-15 1996, San Diego,CA,USA, 1996.

[26] S. Miyazaki, K. Mizukoshi, T. Ueki, T. Sakuma, and Y. Liu, "Fatigue life of Ti-50 at.% Ni and Ti-40Ni-10Cu (at.%) shape memory alloy wires," *Materials Science and Engineering A: Structural Materials: Properties, Microstructure and Processing*
Proceedings of the 1998 International Conference on Martensitic Transformations (ICOMAT 98),Dec 7-Dec 11 1998, vol. A273-275, pp. 658-663, 1999.

[27] P. Nanua, K. J. Waldron, and V. Murthy, "Direct kinematic solution of a Stewart platform," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 438-444, 1990.

[28] C. Pfeiffer, K. DeLaurentis, and C. Mavroidis, "Shape memory alloy actuated robot prostheses: initial experiments," *Proceedings - IEEE International Conference on Robotics and Automation*
Proceedings of the 1999 IEEE International Conference on Robotics and Automation, ICRA99,May 10-May 15 1999, vol. 3, pp. 2385-2391, 1999.

[29] C. Pfeiffer, C. Mavroidis, K. DeLaurentis, and M. Mosley, "Shape memory alloy actuated robot protheses: initial prototypes," *American Society of Mechanical Engineers, Bioengineering Division (Publication) BED*
Advances in Bioengineering - 1999 (The ASME International Mechanical Engineering Congress and Exposition),Nov 14-Nov 19 1999, vol. 43, pp. 145-146, 1999.

[30] T. Raparelli, P. B. Zobel, and F. Durante, "Design of a parallel robot actuated by shape memory alloy wires," *Materials Transactions*, vol. 43, pp. 1015-1022, 2002.

[31] X. Shi and R. G. Fenton, "Solution to the forward instantaneous kinematics for a general 6-DOF Stewart Platform," *Mechanism & Machine Theory*, vol. 27, pp. 251-259, 1992.

[32] H. Tobushi, K. Kimura, T. Sawada, T. Hattori, and P.-H. Lin, "Recovery stress associated with R-phase transformation in TiNi shape memory alloy (properties under constant residual strain)," *JSME International Journal, Series A: Mechanics and Material Engineering*, vol. 37, pp. 138-142, 1994.

[33] F. Wen and C. Liang, "Displacement analysis of the 6-6 Stewart Platform mechanisms," *Mech. Mach. Theory*, vol. v29, pp. 547-557, 1993.

[34] J. Won, K. DeLaurentis, and C. Mavroidis, "Rapid prototyping of robotic systems," presented at ICRA 2000: IEEE International Conference on Robotics and Automation,Apr 24-Apr 28 2000, San Francisco,CA,USA, 2000.

[35] Y. Yam, K.-F. Lei, and P. Baranyi, "Control of a SMA actuated artificial face via neuro-fuzzy techniques," presented at 10th IEEE International Conference on Fuzzy Systems,Dec 2-5 2001, Melbourne,Australia, 2002.

[36] J. P. Yin and C. G. Liang, "Forward displacement analysis of kind of special platform manipulator mechanisms," *Mechanism & Machine Theory*, vol. 29, pp. 1-9, 1994.

[37] J. Yurick T.J., S. E. Mohney, and G. L. Gray, "Shape memory alloy coils optimized for electrical connectors," *IEEE Transactions on Components and Packaging Technologies*, vol. 24, pp. 389-398, 2001.

[38] C. Zhang, R. H. Zee, and P. E. Thoma, "Development of Ni-Ti based shape memory alloys for actuation and control," presented at Proceedings of the 1996 31st Intersociety Energy Conversion Engineering Conference. Part 1 (of 4),Aug 11-16 1996, Washington,DC,USA, 1996.

[39] H. Zou, Q. Wang, Q. Li, and B. Zhang, "Kinematics and workspace analyses of a parallel manipulator for manufacturing," presented at Proceedings of the IEEE International Conference on Industrial Technology,Dec 5-9 1994, Shanghai,China, 1996.